SIEMENS

SIMATIC S5

S5-135U/155U
CPU 928B/CPU 948

Communication

programmable controller

Order No. 6ES5 998-0CN22
Release 01

C79000-H8576-C277-01

Copyright

Disclaimer of liability

We have checked the contents of this manual for agreement with the hardware and software described. Since deviations cannot be precluded entirely, we cannot guarantee full agreement. However, the data in this manual are reviewed regularly and any necessary corrections included in subsequent editions. Suggestions for improvement are welcomed.

Technical data subject to change.

Safety-related guidelines

This manual contains notices which you should observe to ensure your own personal safety, as well as to protect the product and connected equipment. These notices are highlighted in the manual by a warning triangle and are marked as follows according to the level of danger:

Warning
indicates that death, severe personal injury or substantial property damage can result if proper precautions are not taken.

Caution
indicates that minor personal injury or property damage can result if proper precautions are not taken.

Only qualified personnel should be allowed to install and work on this equipment. Qualified persons are defined as persons who are authorized to commission, to ground and to tag equipment, systems and circuits in accordance with established safety practices and standards.

Siemens Aktiengesellschaft

6ES5 998-0CN22
EWK Elektronikwerk Karlsruhe

Printed in the Federal Republic of Germany

Table of Contents

1

Overview

1

Contents of Chapter 1

Overview

1

This chapter discusses the partners with which the CPU 928B/ CPU 948 can communicate, the procedures for transferring data and the requirements for communication. Starting from Section 1.4, the mode of operation for communication is explained in detail. These sections describe how the overall "communication" task is distributed among the individual hardware components of the CPU 928B/ CPU 948 and how communications jobs are processed by the system.

1.1 Communications Partners

The CPU 928B/CPU 948 can communicate with two different partners via two serial interfaces, SI 1 and SI 2 (see Fig. 1-1). If both partners are PGs, then the two interfaces are not completely independent of each other (see CPU 928B and CPU 948 Programming Guides), however, with all other combinations of communications partners, the two interfaces are independent of each other.

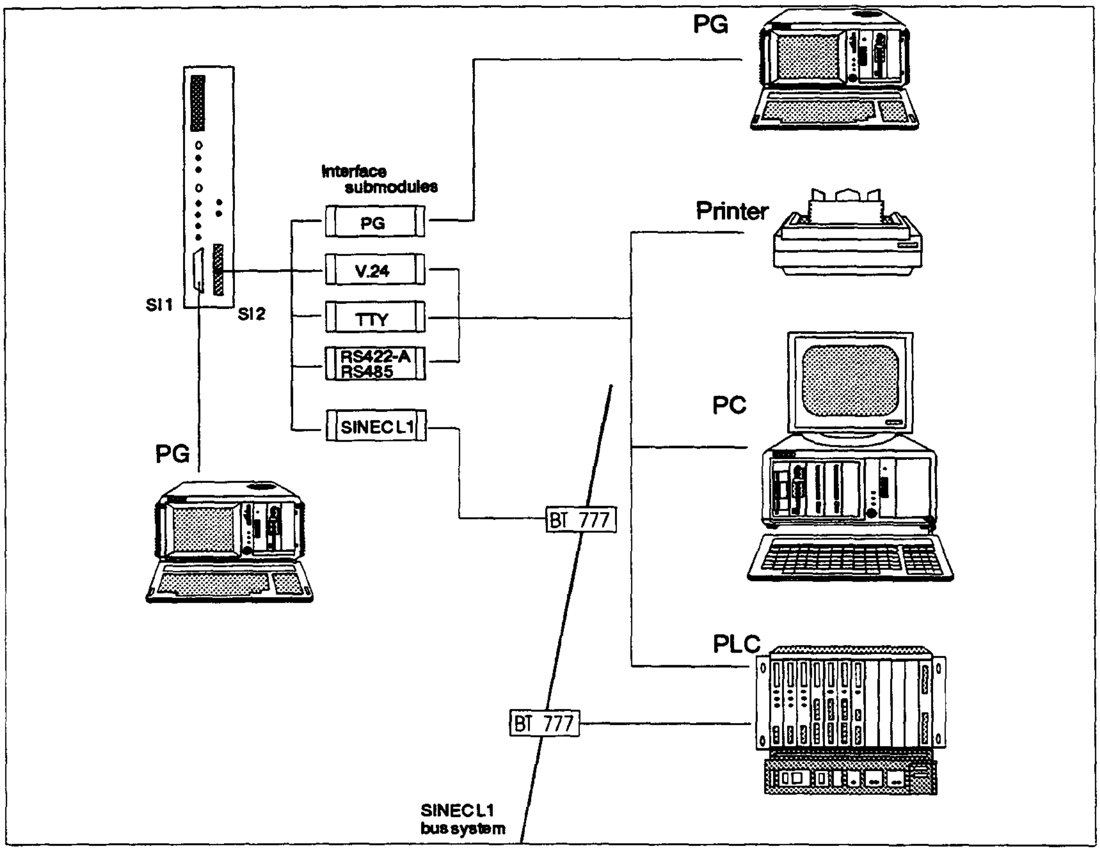


Fig. 1-1 Connecting the CPU 928B or the CPU 948 to communications partners

1

SI 1 is the fixed interface for programmers and operator panels.

SI 2 is a modifiable interface. Its two most important areas of application are communications links and logging. If you use the appropriate interface submodule, you can connect additional units to the CPU 928B/CPU 948 as communications partners. With this interface, the CPU 928B and the CPU 948 extend the range of functions in the S5-135U and S5-155U.

The following figure provides an overview of the possible communications partners.

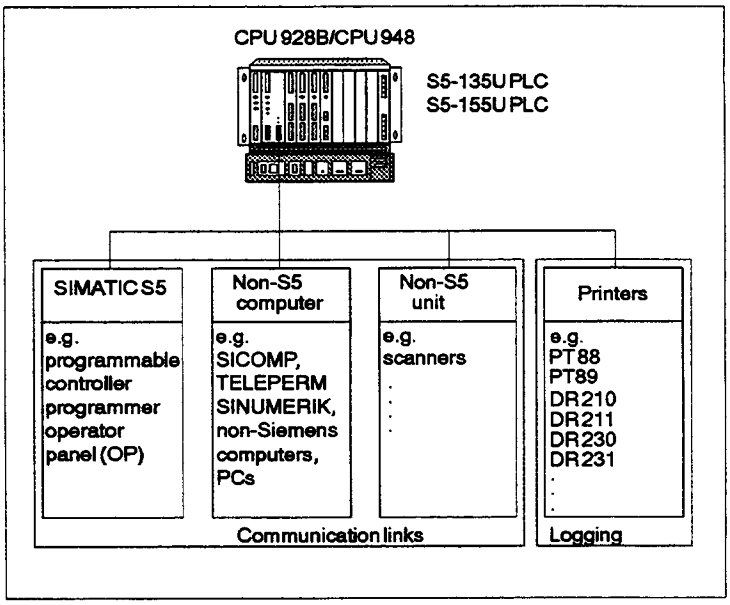


Fig. 1-2 Overview of suitable communications partners

1.2 Transmission Procedures

A connection with asynchronous bit-serial data transmission procedures is possible via the SI 2 interface. The CPU 928B or the CPU 948 support five types of link:

- second PG interface
- RK 512 computer link
- 3964 / 3964R procedure
- "open driver"
- SINEC L1

Second PG interface

This interface is the same in all respects as the first fixed PG interface.

Computer link RK 512

The data transmission between programmable controllers or between a programmable controller and a supervisory non-S5 computer is handled via the SIMATIC S5 standard computer link RK 512. The RK 512 computer link covers three layers of the ISO/OSI 7-layer model (ISO IS 7498):

- the physical layer (layer 1)
- the data link layer (layer 2)
- the transport layer (layer 4)

3964/3964R procedure

The data transmission between a programmable controller and a partner can be handled using the 3964/3964R procedures, providing the PLC and partner operate with the same procedure. Data transmission with the 3964/3964R procedures covers two layers of the ISO/OSI 7-layer model (ISO IS 7498):

- the physical layer (layer 1)
- the data link layer (layer 2)

"Open driver"

The data transmission between a programmable controller and a non-system device with simple transmission procedures is handled by the CPU 928B/CPU 948 using the "open driver". Data transmission with the "open driver" covers one layer of the ISO/OSI 7-layer model (ISO IS 7498):

- the physical layer (layer 1)

SINEC L1

The SINEC L1 is a LAN system for linking programmable controllers to each other and to non-system devices.

1.3 Preparing for Communication

Before you can use the serial interface, you must make certain preparations.

Chapter 2 "Hardware Components" describes, among other things, the following:

- how to install the interface submodule
- the significance and functions of the LED displays on the front panel of the CPU 928B or the CPU 948.

Chapters 3 to 7 describe how you assign parameters for the individual transmission procedures.

1.4 Fundamentals of Communication

1.4.1 Hardware Configuration

To understand how the communications functions are implemented, we must first discuss the hardware of the CPU 928B and the CPU 948.

CPU 928B

The following block diagram shows the hardware components of the CPU 928B that are relevant for communication.

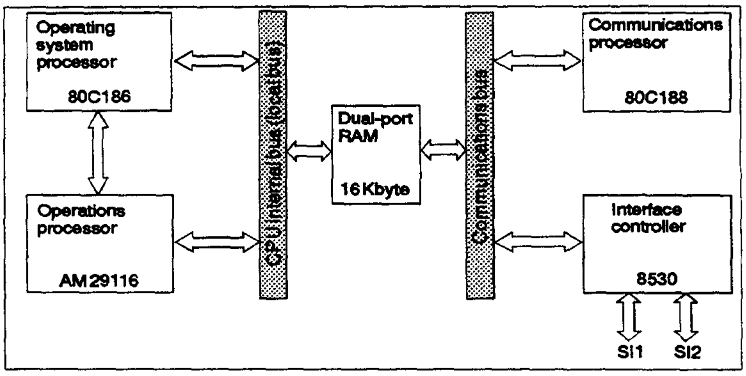


Fig. 1-3 Hardware components of the CPU 928B involved in communication

The components shown in the diagram above perform the following tasks:

- The **operating system processor** handles supervisory tasks such as cycle management and interrupt servicing. It supports the operations processor, transfers communications jobs contained in the user program to the communications processor and transfers the user data between the user program and the dual-port RAM at the request of the communications processor and the user program.

- The **operations processor** processes the STEP 5 operations with the help of a microprogram. It updates the process image and the timers and handles auxiliary functions for the operating system processor. It does not handle any functions directly involved with communication.

- The **dual-port RAM** is a job and data interface between the operating system processor and the communications processor.

- The **communications processor** handles all the communications tasks of the CPU 928B with the outside world. This means that it handles communications functions on the PG interface, the RK 512 computer link, data transmission with the 3964/3964R procedures, with the "open driver" and SINEC L1. The communications processor transfers the user data between the dual-port RAM and the interface controller and carries out front-end processing of the user data.

- The **interface controller** forms the interface between the communications processor and the two serial interfaces SI 1 and SI 2.

CPU 948

The following block diagram shows the hardware components of the CPU 948 that are relevant for communication.

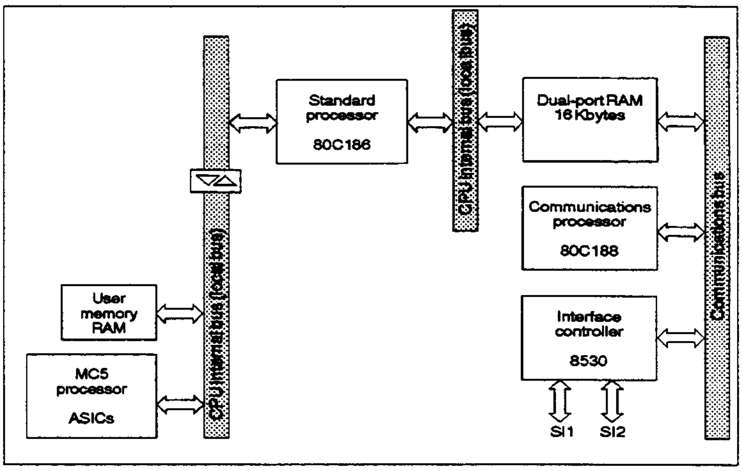


Fig. 1-4 Hardware components of the CPU 948 involved in communication

The components shown in the diagram above perform the following tasks:

- The **MC 5 processor** is not designed to deal actively with communication functions except for error handling (OB 35 call).

- The **standard processor** transfers communication jobs contained in the user program to the communications processor and transfers the user data between the user program and the dual-port RAM at the request of the communications processor and the user program.

- The **communications processor** handles all the communication tasks of the CPU 948 with the outside world. This means that it handles communication functions for the different types of link. The communications processor transfers the user data between the dual-port RAM and the interface controller and carries out front-end processing of the user data.

- The **interface controller** forms the interface between the communications processor and the two serial interfaces SI 1 and SI 2.

1.4.2 System Performance

The communcation on the second serial interface of the CPU 928B/ CPU 948 with the RK 512 computer link, data transmission with the 3964/3964R procedures or with the "open driver" is a pure system activity. This system activity is only invoked when a job is triggered by the user program or communications partner (request by the communications processor).

To achieve user program cycle times that are as far as possible reproducible, the job processing on the operating system processor is divided into processing units. The processing units follow each other at an interval of 10 ms. Each processing unit means that the processing time is extended by approximately 300 µs. If system activities are required simultaneously for a job started by the user program and for a job started by the communications partner, two processing units are inserted at intervals of 10 ms.

CPU 948

Communicating on the second serial interface does not overload the user program. Only the memory access when reading and writing the user data means the processing time is extended by 200 µs per 256 words at intervals of 5 ms.

Jobs triggered by the user program

In contrast to the well-known handling blocks, a communications job is processed asynchronously to the processing of the user program. With the handling blocks, the next operation of the user program is processed before the handling block is terminated. In asynchronous communication on the CPU 928B/CPU 948, a bit is simply set in the send coordination byte (SCB) to start a job (see Section "Send Coordination Byte (SCB)" in the chapter dealing with the appropriate type of link). The system program then processes the job parallel to the user program.

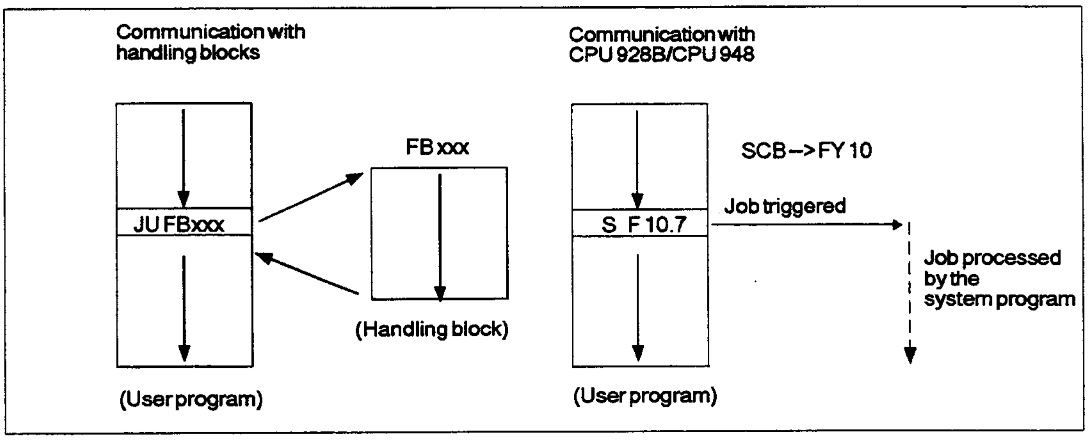


Fig. 1-5 Comparison between communication using handling blocks and via the second serial interface of the CPU 928B/CPU 948

Using the send coordination byte (SCB), both a job to send data (SEND job) and with the RK 512 computer link, a job to fetch data (FETCH job) can be started.

The system program checks at intervals of ≤ 10 ms (CPU 928B) or ≤ 20 ms (CPU 948) whether bit 6 or 7 is set in the send coordination byte (SCB). If one of these bits is set, the job processing is started. The job is processed on the operating system processor/standard processor in two phases.

In the first phase, the parameters are checked and written to the dual-port RAM. Following this, the job is transferred to the communications processor where it is processed. After the job has been executed, the communications processor confirms the completion of the job to the operating system processor/standard processor. In the second phase, the operating system processor/ standard processor then terminates the job.

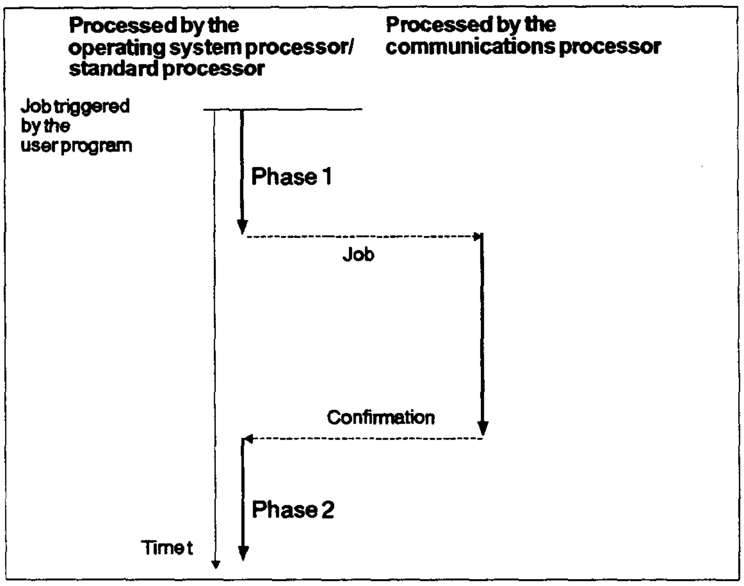


Fig. 1-6 Sequence of a job triggered by the user program

Useful data is transferred between the user memory and the dual-port RAM in phase 1 or phase 2 depending on the job. A maximum of 352 words (CPU 928B) or 2048 words (CPU 948) are transferred per processing unit. For a SEND job, user data are transferred to the dual-port RAM from the user memory in phase 1. For a FETCH job (only RK 512 computer link) user data are transferred from the dual-port RAM to the user memory in phase 2.

Jobs triggered by the communications partner

The system program processes a job triggered by the communications partner in the background. It processes the job parallel to the user program.

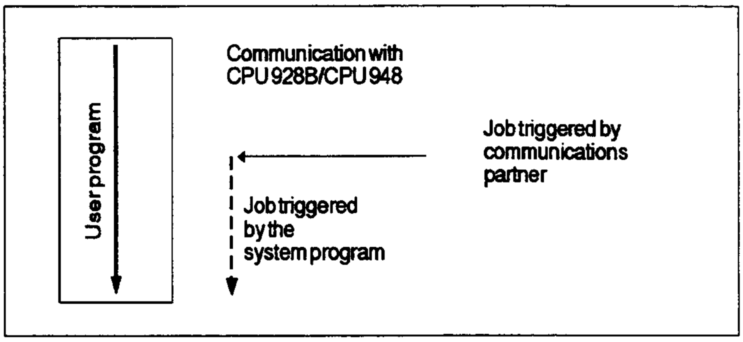


Fig. 1-7 Job processing parallel to user program execution

In this case, the job is triggered for the operating system processor/ standard processor by the communications processor. After the job is triggered, the parameters transferred to the dual-port RAM are checked and the user data transferred from or into the dual-port RAM. A maximum of 352 words (CPU 928B) or 2048 words (CPU 948) are transferred per processing unit. For a SEND job from the communications partner, the user data are transferred from the dual-port RAM to the user memory. For a FETCH job from the communications partner (only RK 512 computer link), the user data are transferred from the user memory to the dual-port RAM.

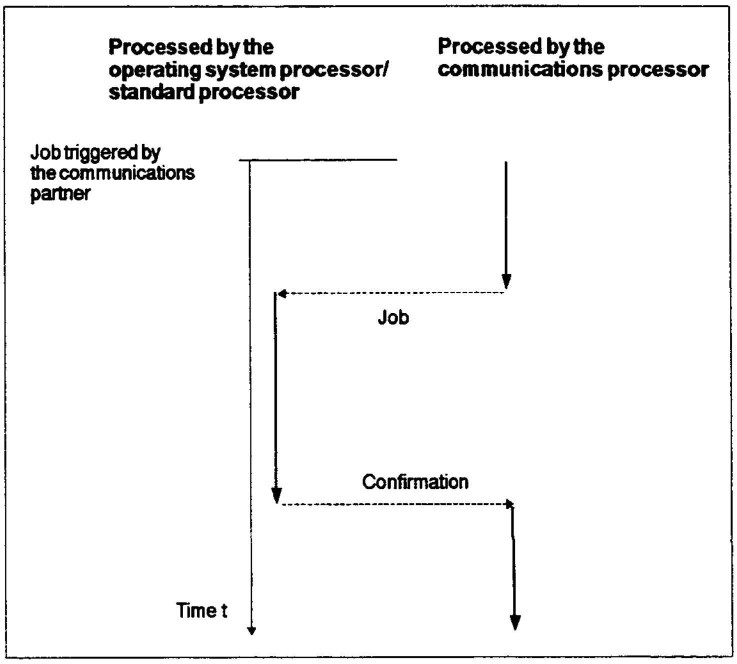


Fig. 1-8 Sequence of a job triggered by the communications partner

1.5 Differences in Communication with the CPU 948 and the CPU 928B

Differences

Communication with the CPU 948 differs from that with the CPU 928B due to the following operating features:

- The CPU 948 can process communication jobs in "soft" stop as well as in the wait mode (program test).
- The CPU 948 can also perform error processing in OB 35 in "soft" stop.
- The system behavior of the CPU 948 cannot be preset in DX 0 if OB 35 is missing. If OB 35 is missing and an error occurs, the CPU 948 remains in the RUN mode and the error numbers are lost.
- In contrast with the CPU 928B, the CPU 948 does not enter the stop mode when parameters are assigned incorrectly in DX 2. The second serial interface is then assigned as a PG interface. There is no response in the operating system date RS3 and RS4 and there is no error message in the ISTACK.
- With the link type "open driver" and SINEC L1, the CPU 948 does not check the following when a job is triggered:
  - whether the transmit mailbox is available complete as specified in DX 2. Instead, it checks whether the length specified in DX 2 for the transmit mailbox lies within the permitted range, whether the number of user data to be transmitted lies within this length range and that the user data to be transmitted actually exist.
  - whether the receive mailbox is available complete as specified in DX 2. Instead, it checks whether the length specified in DX 2 for the receive mailbox lies within the permitted range, whether the number of user data to be received lies within this length range and that the user data to be received actually exist.
- If a mode change is requested by the CPU 948 in the SINEC L1 link type, this is always performed, independent of the mode the CPU is currently in.
  With the CPU 928B the mode change is only triggered if the requested mode is different from the current mode.

2

Hardware Components

2

Contents of Chapter 2

Hardware Components 2

The following safety warning refers to all the hardware components of the CPU 928B and the CPU 948.

Caution

Switch off the central controller before you plug in or remove the module or the memory submodule.

Switch off the central controller before you install or remove an interface submodule.

The basic module and the expansion module of the CPU 928B and of the CPU 948 are one unit. Do not disconnect the mechanical and electrical connections between them.

Avoid touching the components and conductors in any way.

The CPU 928B/CPU948 is supplied without an interface module. The receptacle for the submodule is closed by a plastic cover screwed onto the front panel. Only remove this cover when you want to insert an interface submodule. The CPU 928B/CPU 948 must not be operated with an open submodule receptacle.

2.1 Overview of the CPU 928B and the CPU 948

The S5-135U/155U System Manual contains a product overview, the technical description and the instructions for installation and operation of the CPU 928B/CPU 948. The following chapters describe the functions and operation of the two serial interfaces of the module.

2.1.1 Controls and Displays

All the controls and displays (buttons, switches and LEDs) are on the front panel of the CPU 928B/CPU 948. This front panel is shown in the following figure.

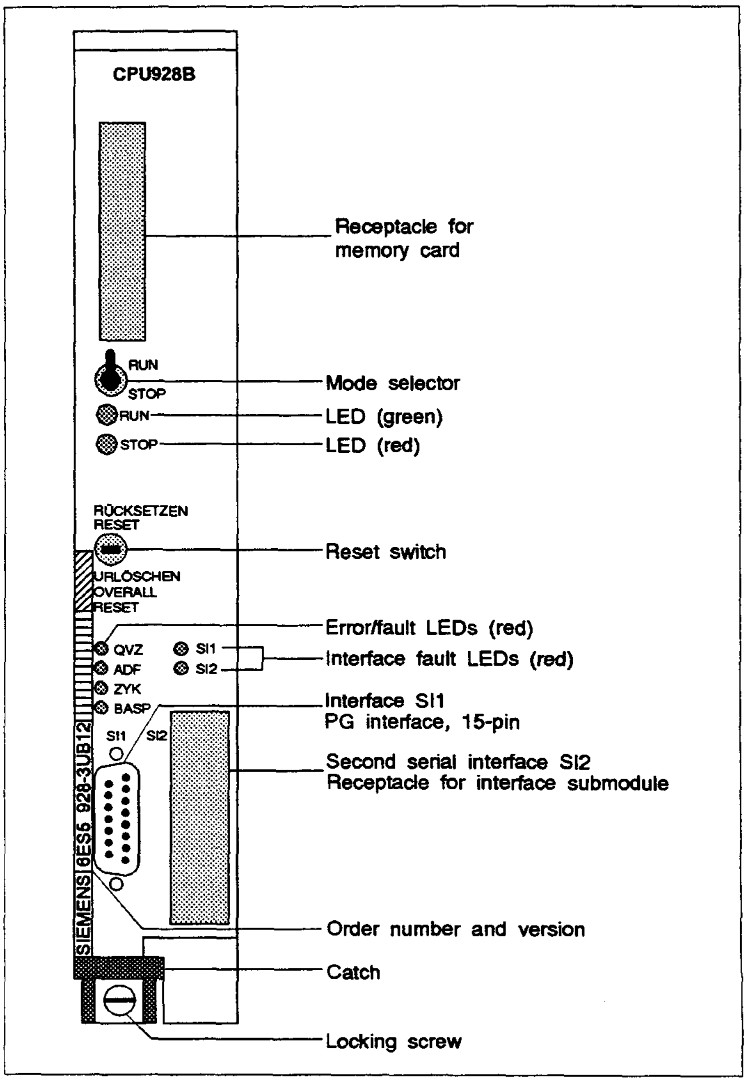


Fig. 2-1 Front panel of the CPU 928B

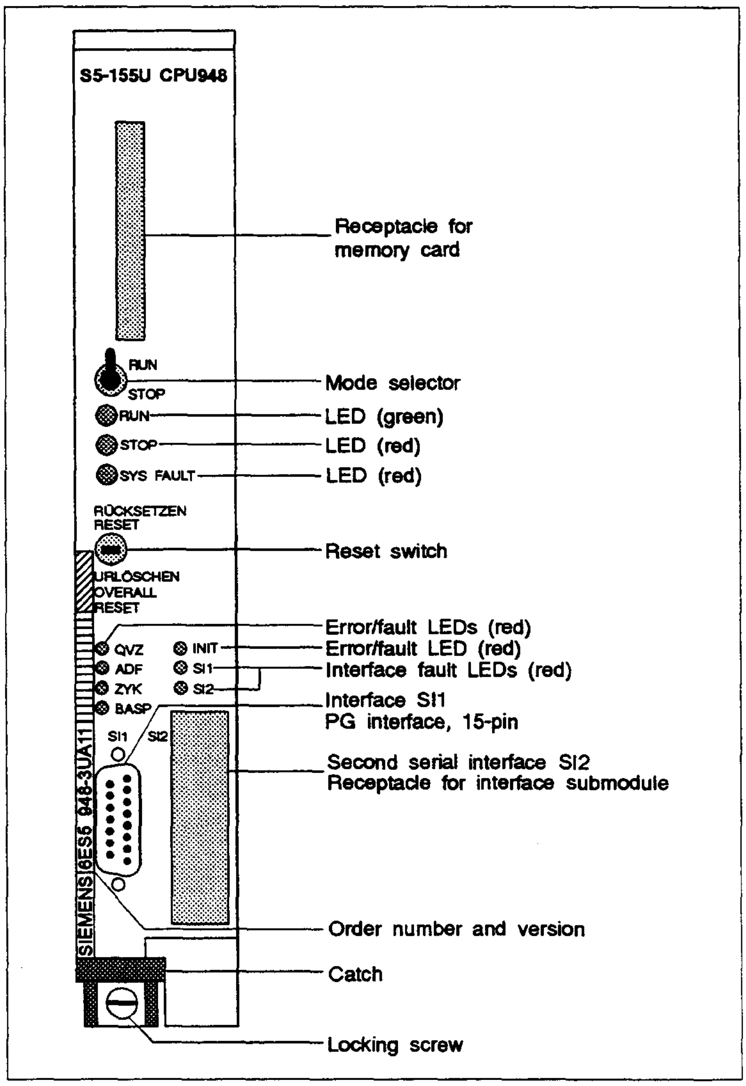


Fig. 2-2 Front panel of the CPU 948

2.1.2 SI 2 Interface

You can use the **second interface** of the CPU 928B and of the CPU 948 in the following ways:

- PG interface (for PG and operator panel)
- interface for the RK 512 computer link
- interface for data transmission using the 3964/3964R procedures
- interface for data transmission with the "open driver"
- interface for data transmission with SINEC L1

Refer to the ordering data in the Appendix for the order numbers of the interface submodules.

If you want to use the second interface as a PG interface, you require the

- **PG submodule**

You also require one of the following interface submodules for the RK 512 computer link, for data transmission using the 3964/3964R procedures and for data transmission with the "open driver":

- **V.24 submodule** (RS 232C)
- **TTY submodule**
- **RS422-A/485 submodule** (only in the RS422-A mode)

For data transmission with SINEC L1 you require the:

- **SINEC L1 submodule**

Caution

The RS 422/485 submodule has a 15-pin front connector the same as the PG/SINEC L1 submodule. If you plug a PG-PLC cable into the submodule instead of the RS 422/485 cable, the submodule may be damaged.
If you connect the RS 422-A/485 cable to a PG interface, the RS 422-A/485 interface on the communications partner may be damaged.

2.1.3 LED Displays SI 1 and SI 2

The two red LED displays SI 1 and SI 2 on the front panel of the CPU 928B or of the CPU 948 indicate, among other things, errors in the parameter assignment during the operation of both serial interfaces and in the operating system of the communication processor. The various meanings of the two LED displays are listed in the following two tables.

The LED display SI 1 belongs to interface SI 1, and the LED display SI 2 belongs to interface SI 2. During the start-up phase following power up, both LEDs are lit for several seconds. SI 1 or SI 2 goes off as soon as its interface is initialized and in normal operation.

Note

If DX 2 is not loaded and there is no submodule installed,
LED SI 2 goes off following the start-up phase and remains off.
This means that the SI 2 interface is not in operation.

The LED displays have different meanings depending on whether you operate SI 2 as a PG interface or as an interface for the RK 512 computer link, for data transmission with the 3964/3964R procedures or for data transmission with the "open driver" or with SINEC L1.

LED displays when SI 1 is used as a PG interface and SI 2 is not used (no submodule installed)

	Interface SI 1 Interface SI 2	Remedy
LED SI 1 ☼ LED SI 2 ☼	**No communication is possible on either interface** Hardware fault or error in the firmware EPROM	 Replace the module
LED SI 1 ☼ LED SI 2 ○	SI 1: **No communication possible** Hardware fault or error in the firmware EPROM SI 2: Interface not used	 Replace the module
LED SI 1 ○ LED SI 2 ☼	SI 1: Interface is initialized and operational SI 2: DX 2 assigned for computer link or Hardware fault or error in the firmware EPROM	 Delete DX 2 Replace the module
LED SI 1 ○ LED SI 2 ○	SI 1: Interface is initialized and operational SI 2: Interface not used	

☼ LED ON ○ LED OFF

LED displays when operating SI 1 and SI 2 as PG interfaces

	Interface SI 1 Interface SI 2	Remedy
LED SI 1 ● LED SI 2 ●	**No communication is possible on either interface** Hardware fault or error in the firmware EPROM	 Replace the module
LED SI 1 ● LED SI 2 ○	SI 1: **No communication possible** Hardware fault or error in the firmware EPROM SI 2: Interface is initialized and operational	Replace the module
LED SI 1 ○ LED SI 2 ●	SI 1: Interface is initialized and operational SI 2: **No communication possible** Wrong submodule plugged in or DX 2 assigned for computer link and PG submodule or no submodule plugged in or Hardware fault or error in the firmware EPROM	Install the PG submodule Assign correct parameters to DX 2 or install the correct submodule Replace the module
LED SI 1 ○ LED SI 2 ○	Both interfaces are initialized and operational	

● LED ON ○ LED OFF

LED displays when operating SI 1 as a PG interface and SI 2 as an interface for the RK 512 computer link, or for data transmission with the 3964/3964R procedures, with the "open driver" or with SINEC L1.

The second interface is only activated as an interface for the RK 512 computer link, or for data transmission with the 3964/3964R procedures, with the "open driver" or with SINEC L1, when

- the appropriate interface submodule is plugged in (not PG submodule)
- DX 2 exists and contains the correct parameters for communication

	Interface SI 1 Interface SI 2	Remedy
LED SI 1 ☼ LED SI 2 ☼	**No communication is possible on either interface** Hardware fault or error in the firmware EPROM	Replace the module
LED SI 1 ☼ LED SI 2 ○	SI 1: **No communication possible** Hardware fault or error in the firmware EPROM SI 2: Interface is initialized and operational	Replace the module
LED SI 1 ○ LED SI 2 ☼	SI 1: PG interface is initialized and operational SI 2: **No communication possible** Wrong or no submodule plugged in or DX 2 with incorrect parameters or Hardware fault or error in the firmware EPROM	Install the correct submodule Enter the correct parameters Replace the module
LED SI 1 ○ LED SI 2 ○	Both interfaces are initialized and operational	

☼ LED ON ○ LED OFF

2.2 Installing and Removing the Interface Submodule

If you wish to plug in an interface submodule, you must first install it in the CPU 928B/CPU 948 before this is installed in the central controller.

Caution

Switch off the power supply to the programmable controller before you remove the CPU 928B/CPU 948.

Installation

Install your interface submodule by following the steps listed below:

Step	Action
1	Check the jumper settings on your interface submodule PG submodule V.24 submodule — See Section 5.8 TTY submodule — in the S5-135U/155U RS422-A/485 submodule — System Manual SINEC L1 submodule When supplied, the jumpers are set so that you can install the interface submodule immediately.
2	Switch the power supply to your programmable controller off.
3	Remove the CPU.
4	Remove the cover of the module receptacle and both screws which hold the interface submodule in place.
5	Plug in the interface submodule through the front panel into the connector (components in the same direction as on the CPU).
6	Secure the submodule with the two screws.
7	Insert the CPU in the central controller.
8	Switch on the power supply to your PLC again.

Removal

To remove your interface submodule, follow the steps outlined below:

Step	Action
1	Switch off the power supply to your programmable controller.
2	Remove the CPU from the central controller.
3	Release the two screws holding the module in place and remove the module from the receptacle.
4	Install a different submodule (as described above) or close the module receptacle with the cover.
5	Insert the CPU in the central controller.
6	Switch the power supply to your PLC on again.

Note

By screwing the interface submodule to the CPU, interference pulses are diverted via the shield ground of the CPU. The CPU must only be operated when the submodule receptacle is closed (cover or submodule inserted).

COM PP Parameter Assignment Software

3

Contents of Chapter 3

3 COM PP Parameter Assignment Software

The COM PP parameter assignment software must be ordered separately.

3.1 What is COM PP and What Functions Does it Provide?

Package

COM PP is supplied on two different diskettes (3 1/2" and 5 1/4") under the operating system PCP/M. Select the diskette to suit your requirements.

You can also use COM PP under the operating system MS-DOS with the aid of the PCP/M emulator (refer to Section 3.2.2).

COM PP consists of the following files:

File name	Content
S5PXCPPX.CMD	COM PP (command file)
S5PDCPPX.DAT	Texts in German
S5PECPPX.DAT	Texts in English
S5PFCPPX.DAT	Texts in French

3.1.1 Description of the Functions

COM PP supports the parameter assignment for a **point-to-point link** for the second serial interface of the CPU 928B or the CPU 948. With it you can do the following:

- create new parameter sets
- modify the parameters of an existing user program or
- print out parameter sets.

Refer to the descriptions of the individual link types for explanations of the parameters.

Creating new parameter sets

COM PP generates a maximum of 3 parameter sets, that are stored in 3 data blocks. The independent parameter sets you create with COM PP are as follows:

- **Basic parameter set in DX2 (root block):**

This contains the identifier for the type of link and references to other data blocks with the

- static parameter set
- dynamic parameter sets
- transmit/job mailbox
- receive mailbox

and the location of the send coordination byte (SCB) and receive coordination byte (RCB).

Static and dynamic parameter sets and the send, job and receive mailboxes should be stored in different data blocks. The defaults in the screen forms support you when you use this strategy.

- **Static parameter set**

This contains all the parameters of the physical and data link layers.

- **Dynamic parameter sets (only for link type RK 512):**

These contain information about the partner station (source or destination) for RK 512 jobs.

Storing data blocks in the user program

The data blocks you program with COM PP are stored in a so-called **program file** (user program). You can select the first 6 characters of the name of the program file, the remainder of the name is always *ST.S5D*, e.g. *TEST01ST.S5D*.

This program file also contains the other STEP 5 blocks. You can decide whether to create function blocks, program blocks etc. with the STEP 5 Basic Package first and then use COM PP or the other way round. You can also store the control program and the link parameters in separate files.

You transfer the data blocks created with COM PP to the CPU 928B or the CPU 948 with the STEP 5 Basic Package.

You can transfer the data created with COM PP and your STEP 5 program to an EPROM submodule using the STEP 5 package EPROM/EEPROM.

Please note the following:

- data block DX 2, the static parameter set and the dynamic parameter sets can be stored and edited with COM PP
- COM PP, however, only assigns the location of the coordination bytes and the send, receive and job mailbox, but does not create them or process their contents (this is performed by the user program).

Modifying parameters

You can modify parameters in an existing user program by selecting the appropriate program file and entering the modified parameters.

When you select the program file a **plausibility check** decides which data in the program file can be programmed by COM PP.

If the data are incorrect, an error message is displayed (refer to Section 3.4). The error message tells you which parameter sets are incorrect. You can then recreate these parameter sets with COM PP.

The plausibility check also takes into account that data blocks not created or modified by COM PP but with the data block editor (DB editor) can also be displayed and modified by COM PP.

Printing parameter sets

You can output generated parameter sets to a printer or to a file.

3.1.2 Working with COM PP

When you work with COM PP you are guided by menus in the **»screen forms.**

Convenient **»help** and **»selection functions** support your programming.

Each input you make is checked. Incorrect keystrokes or entries are detected and the storage of incomplete or syntactically incorrect data is prevented.

In many situations, COM PP provides you with parameter defaults so that you can create a complete program in a few simple steps.

3.2 How to Install COM PP

3.2.1 Making Backup Copies

Please make backup copies of the diskettes you received with the package.

Do not copy the files onto HD diskettes, but onto diskettes with the following quality depending on the size you are using:

Diskette size	Diskette quality
3 1/2"	DS, DD, 135 TPI
5 1/4"	DS, DD, 96 TPI

If you are working with a programmer of the PG 7xx series, use the diskette management program *DISK* to copy the original diskettes. You can call this program from the command line. You can also call *DISK* on a PG 730, PG 750 or PG 770 by pressing function key *F2*, when the following menu is displayed immediately after booting your PG:

F1: Start S5-DOS
F2: Format, duplicate and check diskettes
F6: Load the MS-DOS operating system
F7: Load the FlexOS operating system
F8: Input PCP/M-86 commands

After calling *DISK*, proceed as follows:

1) First format an empty diskette with the option

   - format diskette

2) Then copy the original diskette with the option

   - duplicate diskette.

You are guided through the diskette management program by menus.

On programmers of the PG 6xx series, you can format and copy diskettes using the diskette management program *DSKMAINT*.

3.2.2 Loading and Starting COM PP

COM PP under PCP/M

It is assumed that the PCP/M-86 and S5-DOS operating systems are in user area 0 on the hard disk (as supplied). The files of these operating systems should also have the system attribute (SYS).

COM PP in all USER areas

Loading

If you find it more practical to have COM PP available in various USER areas, copy all the files from the original diskette to USER area 0 and assign the system attribute to the files. You can then work with COM PP in all USER areas.

With the call S5, you load the S5-DOS operating system and the S5-DOS command interpreter (S5-KOMI) is activated. The S5-KOMI displays the packages you can select.

Starting

You call COM PP by positioning the cursor in front of the COM PP package and pressing the function key *F1* (PACKAGE).
The first COM PP screen form then appears.

COM PP in one USER area

Loading

If you want to create program files with COM PP in only one USER area, it makes sense to select an area other than area 1. Copy all the files from the original diskette in PCP/M format to the USER area you have selected.

Starting

Select the hard disk and the USER area containing COM PP at the command level. The call is then the same as when working with COM PP in all USER areas.

COM PP under MS-DOS

If your PG only has one MS-DOS partition, you can use COM PP with the aid of the PCP/M emulator.

Loading

Using the command *PCOPY*, copy all the COM PP files from the original diskette to the directory containing the PCP/M emulator and the STEP 5 Basic Package. If you use STEP 5, stage 6.x, copy the COM PP file using the menu.

Starting

You can call COM PP by positioning the cursor in front of the COM PP package in the S5-DOS command interpreter (S5-KOMI) and pressing *F1* (PACKAGE). The first COM PP screen form then appears, or if you use STEP 5, stage 6.x, call up the COM PP package using the menu.

3.3 How to Work with COM PP

Your dialog with COM PP is based on **»screen forms,** whose content is largely self-explanatory. This section shows you the elements of a screen form and how to enter information in the forms.

A screen form consists of four areas, as follows:

- the overview area
- the programming or configuration area
- the message line and
- the softkey menu

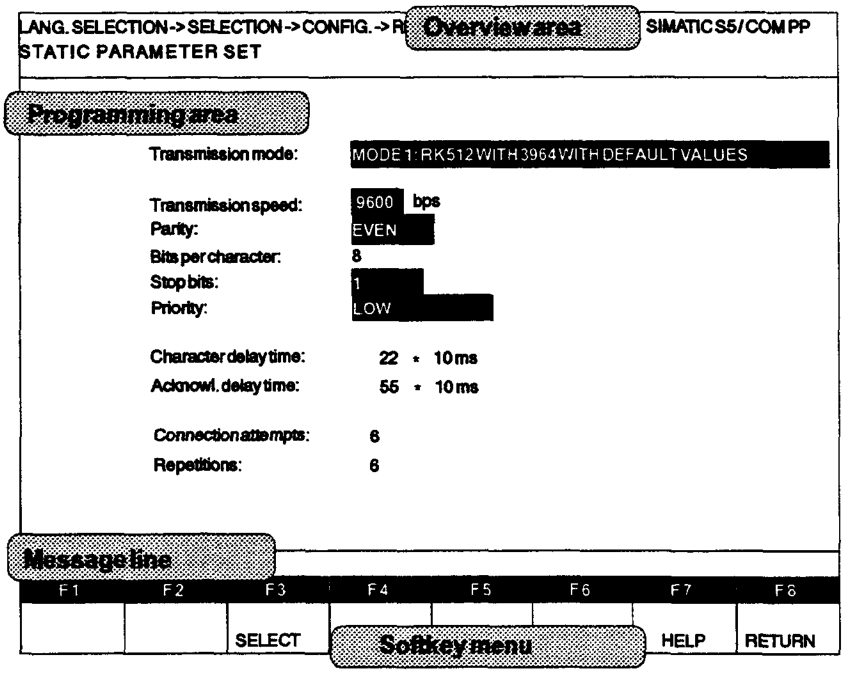


Fig. 3-1 Layout of a screen form

Overview area

This area helps you to see where you are in the program; a path to the current screen form is displayed.

Programming area

This area contains texts that you cannot change and inversely displayed **»input** and **»output fields.** You can position the cursor in input and output fields and modify the displayed parameters or enter new ones.

Message line

The message line displays error and processing messages.

Softkey menu

An important element of COM PP is the **»softkey menu** *F1 to F8*. The softkey menu of a **»screen form** shows you the operations you can carry out with the function keys. Each screen form contains at least one softkey which tells COM PP to display a new screen form.

The ➻screen form hierarchy in the "Reference Section" 3.5 provides you with an overview of all the COM PP screen forms and indicates the screen forms you have to complete to assign parameters.

As necessary, two other screen areas may be displayed:
- the help window and
- the selection window

Help window

When you make inputs, you are supported by the **➻help function.** You can call a help window for each input field.

Position the cursor in an input field and press *F7.* A **➻help window** is now displayed containing information about the input field you selected. The help window is closed again when you have completed the corresponding input field.

Selection window

There are a certain number of possible entries that can be made in an input field. You can call these fixed options by pressing **➻function key** F3. After pressing *F3* either a different string is displayed or if there is a large number of possible entries for an input field, a **➻selection window** is opened.

The top entry in a selection window is selected when the window is opened. Using the **➻cursor control keys**

- cursor up/down

- page up/down

you can select a different entry. You enter your selection by pressing the **➻function key** *F6* (ENTER). Once you enter your selection, the selection window is closed.

Input and cursor control keys

➻ Function keys, ➻cursor control keys and **➻S5 keys** are available.

The significance of the eight function keys *F1* to *F8* is displayed in the softkey menu of the currently displayed screen form. The cursor control keys for positioning the cursor and special S5 keys, for example for calling the **➻help function or** deleting characters are described in Section 3.5 "Reference Section".

Page forwards

Press one of the keys specified in the softkey menu to move on to the next screen form.

Page backwards

Press the **➻function key** *F8*, to return to the previous screen form. In some situations the softkey menu of the current screen form changes to an acknowledgment menu (**➻acknowledgement**) and in the message line you are warned of the risk of losing the data entered in the screen form and prompted to save or abandon your input.

Exiting COM PP

You **➻exit COM PP** and return to S5-KOMI by pressing **➻function key** *F8* repeatedly (**➻return function).** Once again you will be prompted to make certain decisions in acknowledgement menus.

3.4 Which Messages Will You Receive from COM PP?

When necessary, COM PP displays texts belonging to one of the following categories in the message line of a »→**screen form**:

- Errors due to incorrect input (identifier MF)
- Messages (identifier MESS)
- Acknowledgements (identifier ACK)
- Handling errors (identifier ERR)
- Internal errors (identifier IMF)

The texts themselves and the causes of the error are listed below. You are also informed of how to react to the various messages and clear the error.

3.4.1 Errors Due to Incorrect Input

MF.001:	Incorrect key	You have attempted to enter an illegal character in the »→ **input field** marked by the cursor or you pressed the wrong function key.
MF.002:	Incorrect input	The input in the field marked by the cursor is not between the permitted limit values or is not one of the permitted alternatives. Press function key *F7* and read the information about this input field.

3.4.2 Messages

MESS.001:	Incorrect key	You have attempted to enter an illegal character in the → **input field** marked by the cursor or you pressed the wrong function key.
MESS.002:	Incorrect input	The input in the field marked by the cursor is not between the permitted limit values or is not one of the permitted alternatives. Press function key *F7* and read the information about this input field.
MESS.003:	DX2 found	These messages are displayed during the plausibility check. If the message "DX2/static parameter set/dynamic parameter set not found" appears, the program is incomplete. You must then create the missing component.
MESS.004:	DX2 not found	
MESS.005:	Static parameter set found	
MESS.006:	Static parameter set not found	
MESS.007:	Dynamic parameter set found	
MESS.008:	Dynamic parameter set not found	
MESS.009:	New file	This message appears when you select a new file.
MESS.010:	Entered	Your input was checked and written to the external memory.
MESS.011:	Job entered	A job belonging to a *dynamic parameter set* (e.g. send or fetch) was checked and written to the external memory
MESS.012:	Job deleted	A job belonging to a *dynamic parameter set* (e.g. send or fetch) was deleted but not written to the external memory
MESS.013:	Warning! Job faulty	Incorrect jobs have been found in the *dynamic parameter set*. The first error-free job is displayed. Page to job 1 and go through all the jobs step by step. Rewrite incorrect jobs.
MESS.014:	No file of this type found on selected drive	A file of the type *ST.S5D (program file), *DR.INI (printer presets file) or *F1.INI (footer file) does not exist. Either enter a new program file name in the input field or create a new printer presets file or footer file using the STEP 5 package.
MESS.015:	Limit reached	When editing the header or footer for the static parameter set, the limit of 544 characters including control characters and delimiters has been reached.
MESS.016:	Insert function illegal	When editing the header or footer for the static parameter set, it is not possible to insert an extra line.
MESS.017:	Printer not connected or not ready	Either there is no printer connected or the connected printer is not ready or not switched on.
MESS.018:	Active	This message informs you that the PG is active and that you cannot make any entries at the moment. The message appears, e.g. when COM PP is loading a DAT file (texts).
MESS.019:	Printing...	This message appears when the print function is active.

MESS.020:	Printing aborted	This message appears when the print function was aborted with *F2*.
MESS.021:	Printing finished	This message appears when the print function was completed correctly.
MESS.022:	No memory space free	There is not enough memory in the PG or an internal system error has occurred. Exit COM PP and the S5-KOMI, then boot the PG and restart COM PP.
MESS.023:	Space not sufficient (max. DB length exceeded)	When assigning parameters to DX2, note that the job, send and receive mailboxes, the static parameter set, the dynamic parameter set and the coordination byte must be completely within the block boundaries in the DB.
MESS.024:	Coordination bytes: limit value exceeded.	The coordination bytes have different limit values depending on whether they are in a DB, flag or S flag. Press function key *F7* and read the information about this input field.
MESS.025:	Change the entry "Dynamic parameter set:" from NO to YES.	You wanted to assign parameters for a dynamic parameter set. This is, however, only possible if you entered "Dynamic parameter sets: YES" when assigning parameters to DX2.
MESS.026:	Dyn. param. set: last valid job with no. ###!	The maximum number of 255 programmable dynamic parameter sets can also be restricted by the particular location of the dynamic parameter sets, the static parameter set, the coordination flags and the job mailbox. COM PP calculates the last valid job taking into account the parameter assignment of DX2.
MESS.027:	Further jobs not possible acc. to DX2 parameter assignment	The maximum number of 255 programmable dynamic parameter sets can also be restricted by the particular location of the dynamic parameter sets, the static parameter set, the coordination flags and the job mailbox. If you want to program further jobs you must change the parameter assignment for DX2.

3.4.3 Acknowledgments

ACK.001:	Internal system error. Exit?	Exit COM PP by repeatedly pressing function key *F8*. Exit S5 KOMI as well and boot the PG. Then restart COM PP:
ACK.002:	Internal system error. Exit?	
ACK.003:	Screen form modified. Abort latest entry?	This acknowledgment appears when you attempt to exit a screen form before entering your input. Press function key *F1* (YES) if you want to abandon your input.
ACK.004:	Continue?	Read the text in the info window. Press function key *F1* (YES) to execute the function (e.g. ESC or *F8*). Press function key *F3* (NO) and the last input is discarded and COM PP remains in the current screen form.
ACK.005:	Abort printing?	This message appears when you press function key *F2* (STOP PRINT) while printing out.
ACK.006:	Link type changed. Enter?	A different link type from the one currently set is programmed in the .S5D file you have selected. Please remember that a different link type may also require changes or additions to the programmed data (DX2, static parameter set, dynamic parameter sets).
ACK.007:	Link type changed. Stat. param. set must be generated again. Cont.?	DX2 and the static parameter set already existed when the link type was changed. The static parameter set is now no longer suitable for the new link type and must be modified. Press *F1* (YES) to call the screen form for assigning parameters to the static parameter set. Press *F3* (NO) to undo the modification to the link type.
ACK.008:	Stat. param. set longer. Overwrite data behind old stat. p. set?	After the static parameter set (at higher DW addresses) there may be data that have already been created with the DB editor or the COM. If you change the link type, the mode of the static parameter set and the number or length of the header or footer (which together determine the length of the static parameter set), these data may be overwritten. COM PP detects this and displays this message. If you press *F1* (YES) the current parameter set overwrites the existing data in the data block. If you press *F3* (NO) you remain in the current screen form and the data after the old unmodified static parameter set remain unchanged.
ACK.009:	Overwrite non-interpretable data in DB with stat. param. set?	The location you have selected for the static or dynamic parameter set is at an address where there are already data in the data block. These data cannot be interpreted as COM PP parameter sets. If necessary, you can check the DB using the DB editor. If you press *F1* (YES), the current parameter set overwrites the existing non-interpretable data. If you press *F3* (NO) the data are not overwritten.
ACK.010:	Overwrite non-interpretable data in DB with dyn. param. set?	

<table>
<tr><td>ACK.011:</td><td>Link type changed. Overwrite data in DB with stat. param. set?</td><td>A static parameter set already exists. If you press F1 (YES), the old parameter set is overwritten. If you press F3 (NO), COM PP remains in the currently displayed screen form and the parameter set is not stored.</td></tr>
<tr><td>ACK.012:</td><td>Stat. param. set faulty. Must be generated again. Continue?</td><td rowspan="2">The data in the parameter set are corrupted and cannot be interpreted by COM PP. The data block may contain data you created previously with the DB editor or with COM PP.
Using the DB editor, check and if necessary change the location of the parameter set ("from DW") in the parameters in DX2.
If you press F1 (YES) you can create a new parameter set and overwrite the non-interpretable data. If you press F3 (NO), COM PP remains in the currently displayed screen form.</td></tr>
<tr><td>ACK.013:</td><td>Dyn. param. set faulty. Must be generated again. Continue?</td></tr>
<tr><td>ACK.014:</td><td>Link type changed. DX2 must be generated again. Continue?</td><td>A DX2 already exists.
If you press F1 (YES), you can assign new parameters for DX2.
If you press F3 (NO), COM PP remains in the currently displayed screen form.</td></tr>
<tr><td>ACK.015:</td><td>DX2 faulty. Must be generated again. Continue?</td><td>The data from DW 0 onwards of DX2 are incorrect and cannot be interpreted by COM PP: You may have made a mistake when generating DX2 with the DB editor.
Check the DB with the DB editor. If you press F1 (YES) you can create a new DX2 and overwrite the non-interpretable data. If you press F3 (NO), COM PP remains in the currently displayed screen form.</td></tr>
<tr><td>ACK.016:</td><td>Link type changed. Overwrite existing DX2?</td><td>A DX2 already exists. If you press F1 (YES), the old DX2 is overwritten. If you press F3 (NO), COM PP remains in the currently displayed screen form and the old DX2 is retained.</td></tr>
<tr><td>ACK.017:</td><td>Overwrite non-interpretable data in DX2?</td><td>COM PP cannot interpret the data in DX2. Check the DB with the DB editor.
If you press F1 (YES), the DX2 currently displayed by the COM overwrites the existing but non-interpretable DX2.
If you press F3 (NO), COM PP remains in the currently displayed screen form and the old DX2 is retained.</td></tr>
<tr><td>ACK.018:</td><td>Link type changed. Stat. p. set 2 must be generated again. Cont?</td><td>You have changed the link type to "open driver" and the static parameter set, part 2 must be recreated.
If you press F1 (YES), the screen form STAT. PARAMETER SET 2 (PRINTER) appears.
If you press F3 (NO), COM PP remains in the currently displayed screen form.</td></tr>
<tr><td>ACK.019:</td><td>Stat. param. set 2 faulty. Must be generated again. Continue?</td><td>The static parameter set part 2 cannot be interpreted by COM PP. The data block may contain data you created previously with the DB editor or with COM PP.
Check the DB with the DB editor. If you press F1 (YES), you can create a new static parameter set part 2 and overwrite the non-interpretable data. If you press F3 (NO), COM PP remains in the currently displayed screen form.</td></tr>
</table>

ACK.020:	Job faulty. Must be generated again. Continue?	In the dynamic parameter set, you pressed *F1*, *F2* (PAGE BACKW./FORW.) or F4 (GO TO JOB) and COM PP has recognized that the next job is incorrect. If you press *F1* (YES), the screen form for the next job is displayed with the defaults. If you press *F3* (NO), COM PP remains in the current job. You can obtain other jobs by pressing *F4*.
ACK.021:	Quit COM?	With this acknowledgment you indicate whether or not you want to exit COM PP. If you press *F1* (YES), you exit COM PP and return to the S5 KOMI. If you press *F3* (NO), COM PP remains in the currently displayed screen form.
ACK.022:	Stat. param. set 2 not yet in external memory. Quit screen form?	The static parameter set, part 1 is only stored together with the static parameter set 2 on an external data medium (diskette) for the link type "open driver" in the mode "PRINTER OUTPUT". Press *F6* (ENTER) in the STAT. PARAMETER SET 2 (PRINTER) screen form. If you exit the screen form with *F8* (RETURN), the static parameter set part 1 is not written to diskette. If you press *F1* (YES), you exit the screen form and the static parameter set is not entered. If you press *F3* (NO), COM PP remains in the currently displayed screen form.

3.4.4 Handling Errors

ERR.001:	Internal error following read/write access to external memory	Exit COM PP by repeatedly pressing *F8*! Exit S5 KOMI as well and boot your PG. Then start COM PP again. The programmed data may be lost (on diskette or hard disk). For this reason always make a back-up copy.
ERR.002:	External memory (diskette) defective	The diskette drive is open, the diskette is not correctly formatted or defective. Close the drive or format the diskette correctly.
ERR.003:	External memory (diskette) write-protected	The diskette cannot be written to. Remove the write-protection or use a different diskette.
ERR.004:	File is write-protected	You cannot write to the file you have selected. Change the attribute of the file (PCPM: SET XXXXXXST.S5D [RW]) or select a new file.
ERR.005:	File cannot be interpreted (not a STEP 5 file)	The file you have selected does not have a STEP 5 format and cannot be read. Enter a different file name.
ERR.006:	Diskette has been changed	During an editing session with COM PP, the diskette was changed. This is not permitted. Insert the original diskette in the drive again.
ERR.007:	DX2 faulty	This message is displayed during the plausibility check. Set up a new DX2 with COM PP.
ERR.008:	Printer parameters cannot be read	An error has occurred reading the printer file or the printer file does not exist. Set up a new printer file with the STEP 5 basic package keeping to the name convention *DR.INI.
ERR.009:	Print to file not possible	There is not enough memory for the print to file function. Create space on the PG by deleting unnecessary files.
ERR.010:	Static parameter set faulty	These messages appear during the plausibility check. Set up new parameter sets with COM PP.
ERR.011:	Dynamic parameter set faulty	
ERR.012:	External memory (diskette) full	There is not enough space on the diskette and the data cannot be stored. Exit COM PP and delete the files you no longer require.
ERR.013:	Directory full	No more files can be created because the maximum number of files has been reached. Exit COM PP and delete the files you no longer require.
ERR.014:	Internal system fault	The file you have selected does not exist (e.g. footer file).
ERR.015:	File too big	The file cannot be read into COM PP.
ERR.016:	Illegal ack. delay time (HELP: limit values. SELECT: min value)	The limit value for the acknowledgment delay time has been exceeded. Remember that the minimum values vary with the transmission speed. Press *F7*, to display the limit values or *F3*, to obtain the minimum value suitable for the transmission speed.

ERR.017:	Illegal char. delay time (HELP: limit values. SELECT: min value)	The limit value for the character delay time has been exceeded. Remember that the minimum values vary with the transmission speed. Press *F7*, to display the limit values or *F3*, to obtain the minimum value suitable for the transmission speed.
ERR.018:	Drive not available	You have selected a non-existent drive.
ERR.019:	Static parameter set 1 faulty	These messages appear during the plausibility check. Set up the appropriate new parameter set.
ERR.020:	Static parameter set 1 too short. Cannot be interpreted	
ERR.021:	Static parameter set 2 faulty	
ERR.022:	Static parameter set 2 too short. Cannot be interpreted	
ERR.023:	Parameter set too long (max. DB address DW 2042 exceeded)	The parameter set cannot be stored. Change the mode of the static parameter set (select shorter mode) or shorten the header/footer or change your entry for "from DW" in the parameters for DX2.
ERR.024:	No. of header/footer lines illegal (cannot be processed)	The header/footer editor cannot interpret the header/footer of the static parameter set (number of lines and delimiters do not correspond to the text).
ERR.025:	DX2 too short. Cannot be interpreted	These messages appear during the plausibility check. Set up the appropriate new parameter set.
ERR.026:	DX2 faulty	
ERR.027:	Stat. param. set too short. Cannot be interpreted	
ERR.028:	Dyn. param. set too short. Cannot be interpreted	
ERR.029:	No. of header/footer lines in static parameter set 2 illegal	

ERR.030:	Static parameter set and coordination bytes overlap	When you assigned parameters to DX2, you specified the location of the job, transmit and receive mailboxes so that at least two memory areas are in the same data block and overlap each other. Change the parameters in DX2
ERR.031:	Static parameter set and dynamic parameter set overlap	
ERR.032:	Static parameter set and job mailbox overlap	
ERR.033:	Dynamic parameter set and job mailbox overlap	
ERR.034:	Dynamic parameter set and coordination bytes overlap	
ERR.035:	Job mailbox and coordination bytes overlap	
ERR.036:	Static parameter set and transmit mailbox overlap	
ERR.037:	Static parameter set and receive mailbox overlap	
ERR.038:	Transmit mailbox and receive mailbox overlap	
ERR.039:	Transmit mailbox and coordination bytes overlap	
ERR.040:	Receive mailbox and coordination bytes overlap	
ERR.041:	Error reading external memory. Plausibility check not done.	When you exit a user program (return to the screen form "SELECTION") a plausibility check is made on the parameter sets, referenced in DX2. This means that all the data blocks involved must be read from external memory (diskette). If an error occurs when reading from diskette, (e.g. diskette drive opened accidentally) the plausibility check cannot be performed.
ERR.042:	Stat. param. set overlaps mailboxes or coordination bytes	If the static parameter set and the data programmed in DX2 are in the same data block the data may be overlapped if the static parameter set is extended. Change the location of the mailboxes, coordination bytes and parameter sets.
ERR.043:	Internal system error	Exit COM PP by repeatedly pressing *F8*! Exit S5 KOMI as well and boot your PG. Then start COM PP again.

3.4.5 Internal Errors

IMF.001:	Screen form not found	Exit COM PP by repeatedly pressing *F8*! The programmed data may be lost (on diskette or hard disk). For this reason always make a back-up copy
IMF.002:	Field not found	
IMF.003:	Illegal MAINT call	

3.5 Reference Section

A

Acknowledgement

COM PP prompts you to acknowledge or confirm certain intentions and system messages, e.g. to avoid losing data. These prompts and system messages appear in the message line of the current **»screen form.** The softkey menu also changes to an **acknowledgement menu.**

ACK. 003: Screen form modified. Abort latest entry?							
F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
YES		NO					

Fig. 3-2 Acknowledgement menu

The acknowledgement menu only displays the **»function keys**

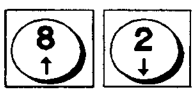
- F1: YES and
- F3: NO

If you press *F1,* the action in the message line is executed. If you press *F3,* the action in the message line is not executed and the previous status is adopted again.

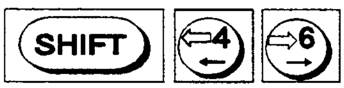
C

Cursor control keys

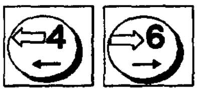
The description of the keys is based on the programmers of the PG 7xx series. For the key assignments of the programmers of the PG 6xx series, please refer to the appropriate manuals.



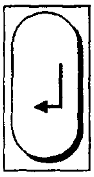
CURSOR up/down
The cursor is positioned in the higher or lower **»Input field.**



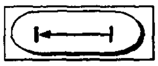
SHIFT CURSOR left/right
The cursor is positioned in the previous or last **»Input field.**



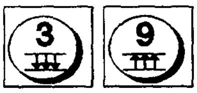
CURSOR left/right
The cursor is positioned on the previous/next character in the input field. If the limit of the field is reached the cursor moves to the next or previous field.



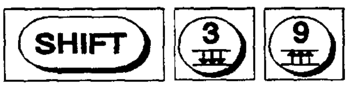
RETURN KEY
Selects an entry in a selection window and closes the window.
Positions the cursor in the next input field.
(The cursor jumps from the last field of a screen form to the first field).
Begins a new line in an editing field.



BACKSPACE
Deletes the character before the current cursor position.



PAGE forwards/backwards
You can page forwards or backwards. If there are not enough data for a further page, the screen scrolls to the end. You can use these keys in ➻**selection windows.**



You jump to the start/end of a ➻**selection window.**

E

Enter

You can *enter* your input by pressing *F6*. The key *F6* has the following functions depending on when it is pressed.

- Entering a screen form stores the data that have been input and branches to the next screen form.
- In a ➻**selection window** the selected entry is entered in the corresponding ➻**input field** and the selection window is closed. You can also enter a selection from a selection window by pressing the RETURN key (➻**cursor control keys**).

Exiting COM PP

You exit COM PP and return to the S5 KOMI by pressing function key *F8* repeatedly (➻**return function**). You will be prompted to make certain decisions (➻**acknowledgement**).

F

Function keys

The eight function keys *F1* to *F8* are assigned a function in the sofkey menu of the currently selected **»screen form.** Keys *F3*, *F6*, *F7* and *F8* usually have the following functions:

F3	SELECT opens a **»selection window.**
F6	ENTER (same as enter key): closes a **»selection window,** enters the data you have input and displays the next screen form.
F7	HELP (same as HELP key): triggers the **»help function** and the opens a **»help window.**
F8	RETURN (same as ESC key): exits the screen form, returns to the next higher level in the **»screen form hierarchy.**

H

Help function

A help function is available for all ➻**input fields**. This is called with function key F7. A ➻**help window** is then opened which contains information about the required input.

3

Help window

A *help window* displays a help text to explain the meaning of an ➻**input field** and the possible entries.

A help window remains open while you edit the corresponding input field. It is closed when you exit the field.

It is not possible to display a help and selection window at the same time. You must first close a selection window before calling the help function.

I

Input field

Input fields are in the programming area of a »→**screen form** and are displayed inversely. You can also recognize an input field since you can position the cursor in it.

Editing

The following keys are available for editing an input field:

- alphanumeric keys
- »→**cursor control keys** (CURSOR left/right, BACKSPACE)
- »→**S5 keys** (DELETE, EXPAND HORIZONTALLY)

Completing

You can complete an input field by

- editing or
- using the »→**selection function**.

Selection

When there are several input fields in the programming area of a screen form, you can exit an input field and select a new field in the following ways:

- »→**cursor control keys**
  (CURSOR up/down
  CURSOR to previous/next input field,
  CURSOR left/right,
  RETURN) or
- by filling out the field completely.

O

Output field

An output field can be displayed either in the programming area of a ➸**screen form** or in a ➸**selection window**. Data entered previously or those available for selection are displayed.

3

P

PRINT

You can output parameter sets to a printer or file. The printout has a header and footer. The header cannot be modified, the footer is determined by a footer file. You can specify this footer file and a printer file in the PRINT screen form (➸**screen form hierarchy**). The creation of footer and printer files is not part of COM PP, but they can be created in the SIMATIC utilities FOOTER and PRINTER.

R

Return function

You trigger the *return function* by pressing ➸**function key** *F8*.

The return function closes a ➸**selection window** without any changes being made.

If there is no selection window open, the return function calls the next higher ➸**screen form** in the hierarchy. Prior to this an ➸**acknowledgement** may be prompted to warn you about possible loss of data.

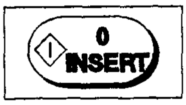
S

S5 keys

The description of the keys is based on the programmers of the PG 7xx series. For the key assignments of the programmers of the PG 6xx series, please refer to the appropriate manuals.

DELETE
Deletes the character at the current cursor position

EXPAND HORIZONTALLY
Inserts a blank at the current cursor position



ENTER KEY
This key enters your input
and has the same function as ➻**function key** *F6*.

ESCAPE KEY
With certain restrictions this key has basically the same functions as ➻**function key** *F8*. (You cannot exit acknowledgment menus with the *ESC* key.)

HELP KEY:
triggers the ➻**help function** and opens a ➻**help window**.

Screen form

Screen forms are the user interface of COM PP. A screen form is divided into 25 lines each with 80 characters and is made up of four areas.

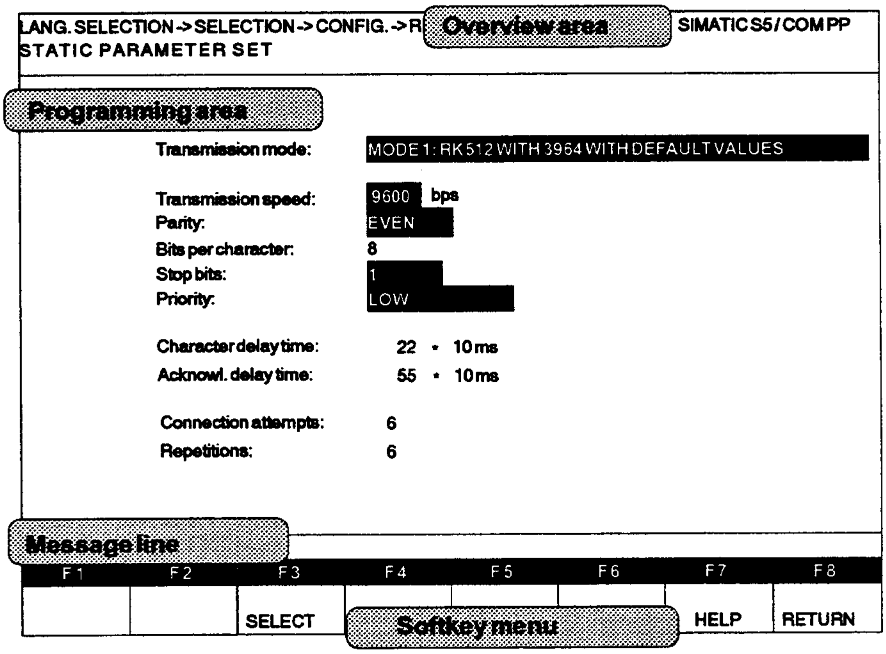


Fig. 3-3 Areas of a screen form

- The **overview area** displays the path to the current screen form in the 1st line and the name of the screen form in the 2nd line.
- The programming area can contain static texts and **»input** and **»output fields**. This area has 18 lines.
- The **message line** displays messages (refer to Section 3.4). This is the last line before the sofkey menu.
- The **softkey menu** indicates the meaning of the function keys. It takes up the last three lines of the screen.

Screen form hierarchy

The individual »screen forms are grouped into various hierarchical levels according to their function.

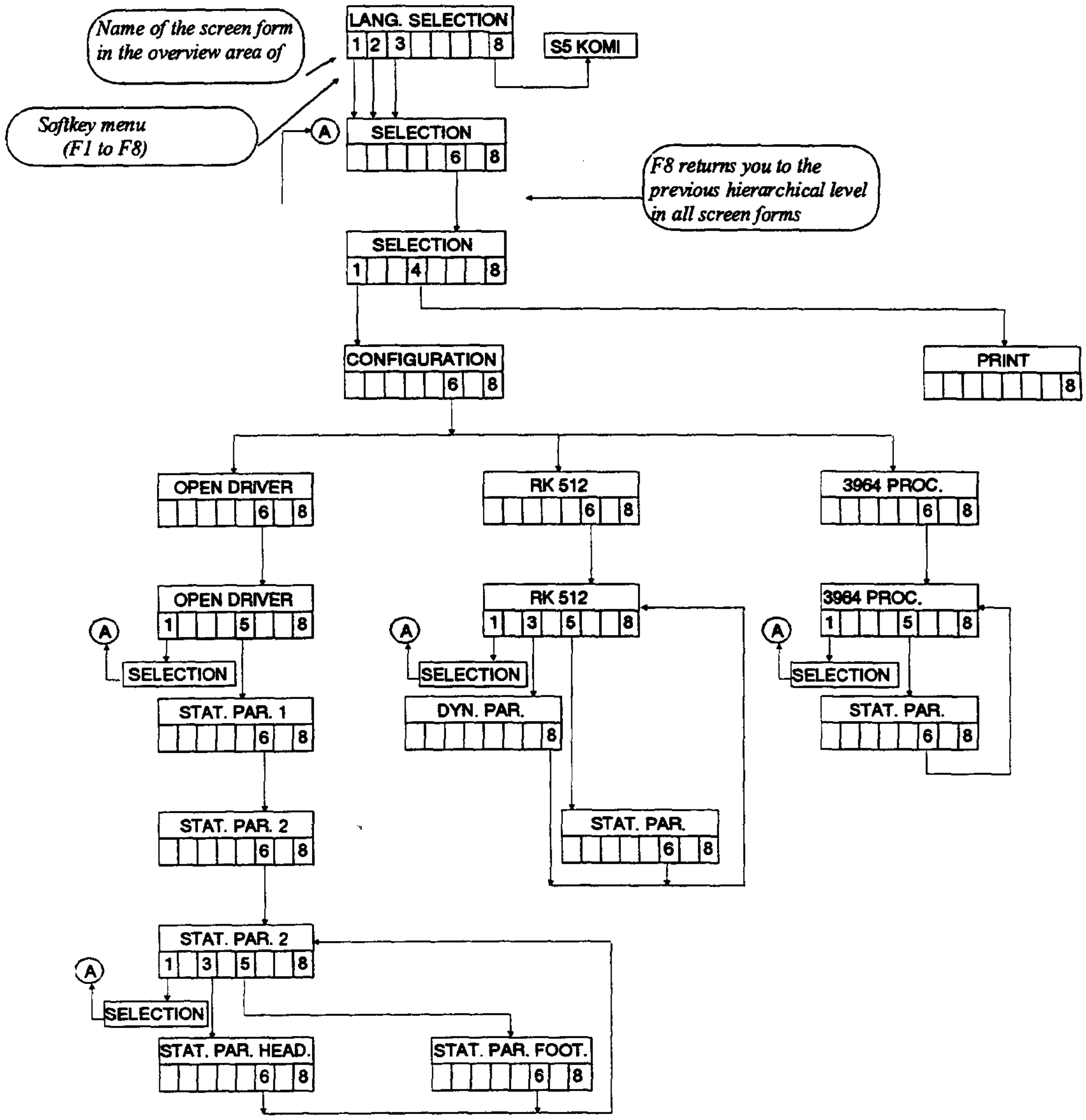


Fig. 3-4 Screen form hierarchy

Selection function

This function is used to enter one of the options in the ➻**input field** marked by the cursor. You call the selection function by pressing ➻**function key** *F3*.

The function allows selections to be made in two ways:

- If there is only a small number of possible entries a different entry is displayed in the input field each time *F3* is pressed.
- If there are numerous possible entries, a ➻**selection window** appears, from which you can select one of the displayed entries.

Selection window

If you start the ➻**selection function** in an input field with a large number of possible entries, a selection window appears on the screen. These windows list all the elements available (e.g. program files on a drive). Twelve output lines are available for the elements.

The first element in the list is always selected initially and displayed inversely. You can select a different element using the ➻**cursor control keys**.

If there are more elements than output lines you can also scroll with the cursor up/down keys. With the page forwards/backwards keys you can also page through the list. If there are elements above or below the currently displayed section this is indicated by an arrow in the appropriate direction.

Once you have found and selected the element you require you can enter it in the input field by pressing ➻**enter**. The selection window is then closed.

The ➻**return function** closes a selection window without entering the data.

It is not possible to display a selection and help window at the same time. Calling the selection function closes a help window and opens a selection window.

RK 512 Computer Link

4

4

Contents of Chapter 4

RK 512 Computer Link

4

4

You can use the second interface of the CPU 928B/CPU 948 as an interface for the

- **SIMATIC S5 RK 512 standard computer link**
  with the 3964 or 3964R procedure.

The RK 512 computer link of the CPU 928B/CPU 948 uses the identical message structure as the CP 524 and CP 525, CP 544 communications processors and the IM 512C interface module.

4.1 Introduction to RK 512

Using the RK 512 computer link, data can be exchanged between two communications partners via a point-to-point link.

The SIMATIC S5 RK 512 computer link ensures a high degree of transmission reliability, since the RK 512 protocol includes several layers of the ISO/OSI layer model (ISO IS 7498):

- The physical layer (layer 1):
  this layer stipulates the physical transmission of the data bytes (physical characteristics of the connection, transmission speed, ...)

- The data link layer (layer 2):
  the data bytes are transmitted using the 3964 or 3964R transmission procedure. This adds start and end characters to the data bytes and initiates repetitions if errors occur.

- Network layer (layer 3):
  this does not exist in the RK 512, since it is a pure point-to-point link.

- Transport layer (layer 4):
  the RK 512 replies to every correctly received command message with a reply message (see Section 4.6.3). This allows the sender to check that its data have arrived completely at the partner or whether the requested data are available at the partner.

This introduction contains basic information about the RK 512 computer link, as follows:

- application
- jobs
- user program

4

4.1.1 Application

The RK 512 computer link implements the direct data exchange between the CPU 928B or the CPU 948 in an S5-135U or S5-155U and one of the following partners:

- **another CPU 928B or CPU 948** in the S5-135U and S5-155U programmable controllers
- **a CP 544** in the S5-115U, S5-115H, S5-135U, S5-155U and S5-155H programmable controllers
- **a CP 525** or **CP 524** in the S5-115U, S5-115H, S5-135U, S5-150U, S5-155U and S5-155H programmable controllers
- an **IM 512** interface module in the S5-110S, S5-130W, S5-150A/K, S5-150S and S5-150U programmable controllers
- a **non-SIMATIC computer** capable of handling RK 512 (e.g. SICOMP M, SICOMP R)

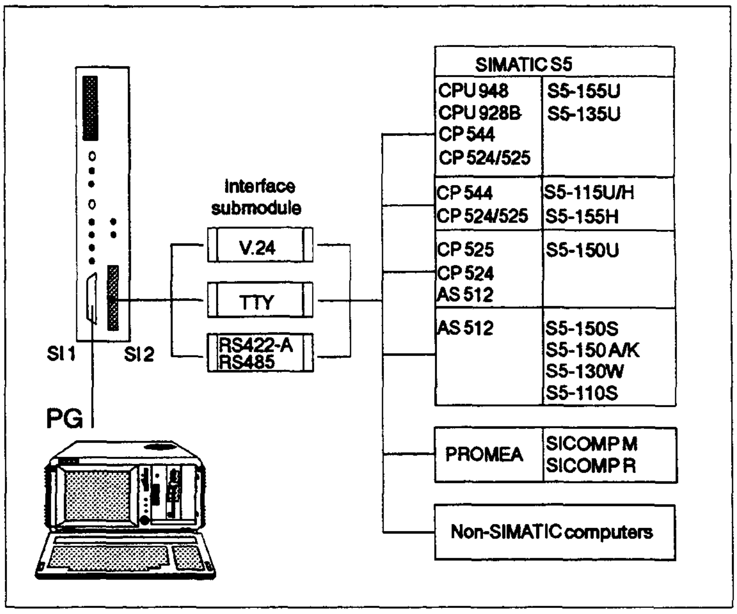


Fig. 4-1 Data exchange

Various interface submodules are available for the RK 512 computer link, as follows:

- **V.24 submodule**
- **TTY submodule**
- **RS422-A/485 submodules** (only in the RS422-A mode).

Install the submodule you require for your application in the submodule receptacle of the second interface of the CPU 928B or the CPU 948 (see Chapter 2, "Hardware Components"). You establish the link to the partner using a standard connecting cable. If you cannot use the standard connecting cables, and prepare your own cable, make sure that you keep to the pin assignments explained in Section 5.8 of the S5-135U/155U System Manual.

Here you will also find additional information on the interface submodules and the standard connecting cables.

Set the transmission speed for the data exchange as suitable for the communications partner (see Section 4.2.2). The following transmission speeds are possible:

- 110 bps to 19200 bps with the RS422-A/485 submodule
- 110 bps to 19200 bps with the V.24 submodule
- 110 bps to 9600 bps with the TTY submodule.

4.1.2 Jobs

With the SIMATIC S5 RK 512 standard computer link you can implement the following active jobs:

- send data: **SEND job**
- fetch data: **FETCH job**

With the SIMATIC S5 RK 512 standard computer link you can implement the following passive jobs:

- receive data: **SEND job on the partner**
- prepare data to be fetched: **FETCH job on the partner**

A SEND job and a SEND job on the partner form a pair. A SEND job on the CPU 928B/CPU 948 leads to a SEND job on the partner and vice versa.

A FETCH job and a FETCH job on the partner form a pair. A FETCH job on the CPU 928B/CPU 948 leads to a FETCH job on the partner and vice versa.

SEND job

The CPU 928B and the CPU 948 are active, they send data to the partner. The partner is passive - it receives data.

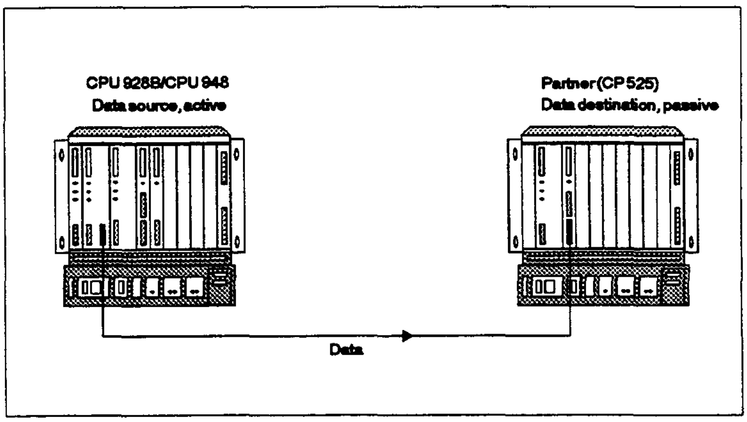


Fig. 4-2 Example of a SEND job with a CP 525 as communications partner

FETCH job

The CPU 928B and the CPU 948 are active - they fetch data from the partner. The partner is passive.

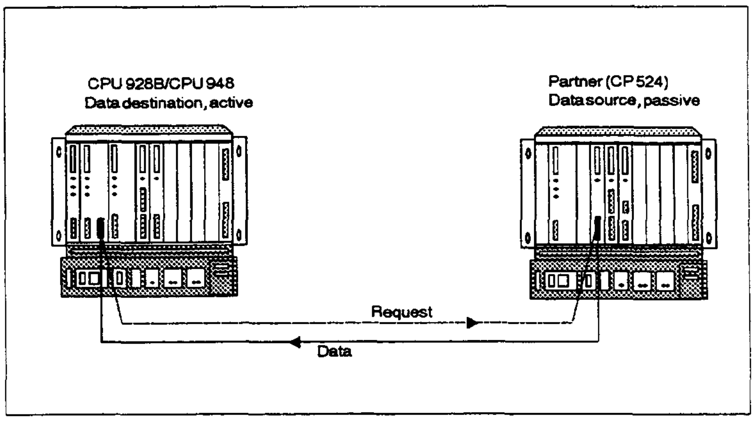


Fig. 4-3 Example of a FETCH job with a CP 524 as communications partner

SEND job on partner

The partner is active - it sends data to the CPU 928B or the CPU 948. The CPU 928B/CPU 948 is passive - it receives data.

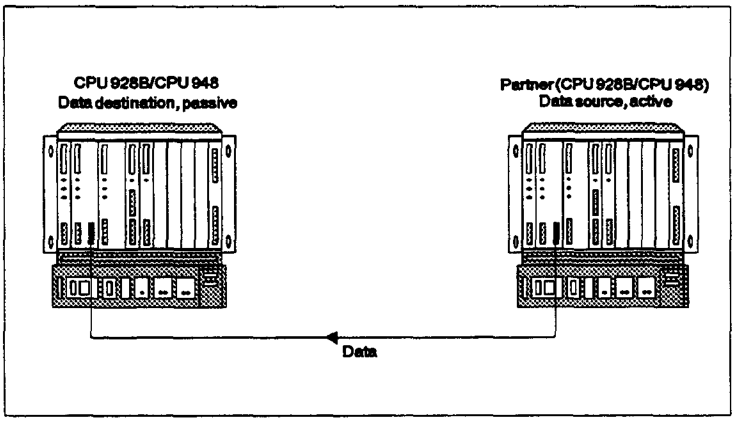


Fig. 4-4 Example of a SEND job on the partner with a second CPU 928B or CPU 948 as communications partner

Note

For the CPU 928B the following applies:
If follow-on messages are received during a partner SEND job while the CPU 928B is in the STOP mode, the last follow-on message is acknowledged with error 2AH in the reply message.

FETCH job on partner

The partner is active - it fetches data from the CPU 928B/CPU 948. The CPU 928B/CPU 948 is passive.

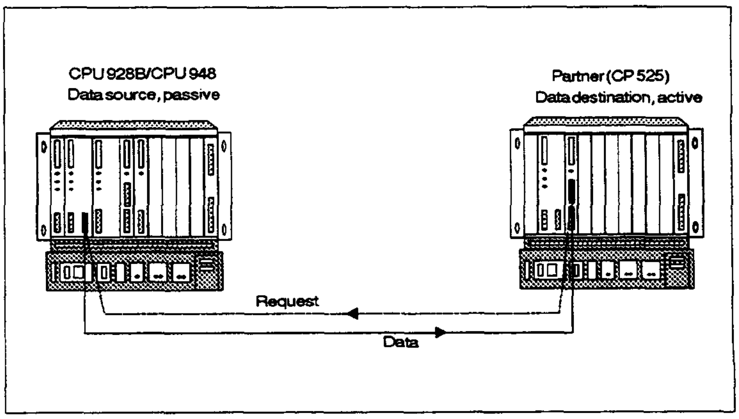


Fig. 4-5 Example of a FETCH job on the partner with a CP 525 as communications partner.

Note

In contrast to the CP 525, CP 524, CP 544 and IM 512, you can only transmit data words from a data block (DB) or an extended data block (DX). Per job, a maximum of 2048 data words are possible. You cannot transmit, fetch or receive inputs/outputs, flags, timers, counters etc.

4.1.3 Incorporating RK 512 into the User Program

This section provides you with an overview of how to incorporate the RK 512 computer link into a user program.

When creating your user program, you should follow the steps explained below:

Prior considerations

- Which devices will the CPU 928B or the CPU 948 exchange data with?
- Which interface submodule do you require?
- Which data will be exchanged?
- What will you call the jobs?

Assigning parameters to the interface

You will require the DX 2, a static parameter set, dynamic parameter sets, a job mailbox and a send coordination byte (SCB).

DX 2

You must enter the following information in **DX 2**:

- link type: RK 512 computer link
- location of the static parameter set
- location of the first dynamic parameter set
- location of the job mailbox
- location of the send coordination byte (SCB)

Static parameter set

In the DB/DX containing the static parameter set you must enter the parameters for the physical layer and for the data link layer.

The parameters for the physical layer (layer 1) are as follows:

- transmission speed (bps)
- parity
- bits per character
- stop bits

The parameters of the data link layer (layer 2), that determine the characteristics of the 3964 or 3964R transmission procedure, are as follows:

- mode: 3964/3964R procedure with defaults or assigned values
- priority
- character delay time
- acknowledgement delay time
- connection attempts
- number of repetitions

Dynamic parameter sets

In the DB/DX containing the dynamic parameter sets you must enter the job parameters that involve the partner:

- SEND or FETCH job
- data destination in the partner for a SEND job or data source in the partner for a FETCH job
- coordination flags (byte and bit number)
- number of the CPU when you exchange data via a CP 522, CP 525 or CP 544 with several CPUs in the partner.

Job mailbox

In the DB/DX containing the job mailbox, you must enter the job parameters for your own CPU 928B/CPU 948:

- SEND or FETCH job
- job number (1 to 255) to select the required dynamic parameter set
- location of the condition codeword
- data source for a SEND job or data destination for a FETCH job
- length of the data to be transmitted
- location of the parameter assignment error byte (PAFE)

SCB

With the send coordination byte (SCB) you can perform the following:

- start a SEND job
- start a FETCH job
- reset the RK 512 computer link

You can generate the parameter set in DX2, the static parameter set and the dynamic parameter sets either via the **DB editor** or via the menus in the **COM PP Parameter Assignment Software** (see Chapter 3).

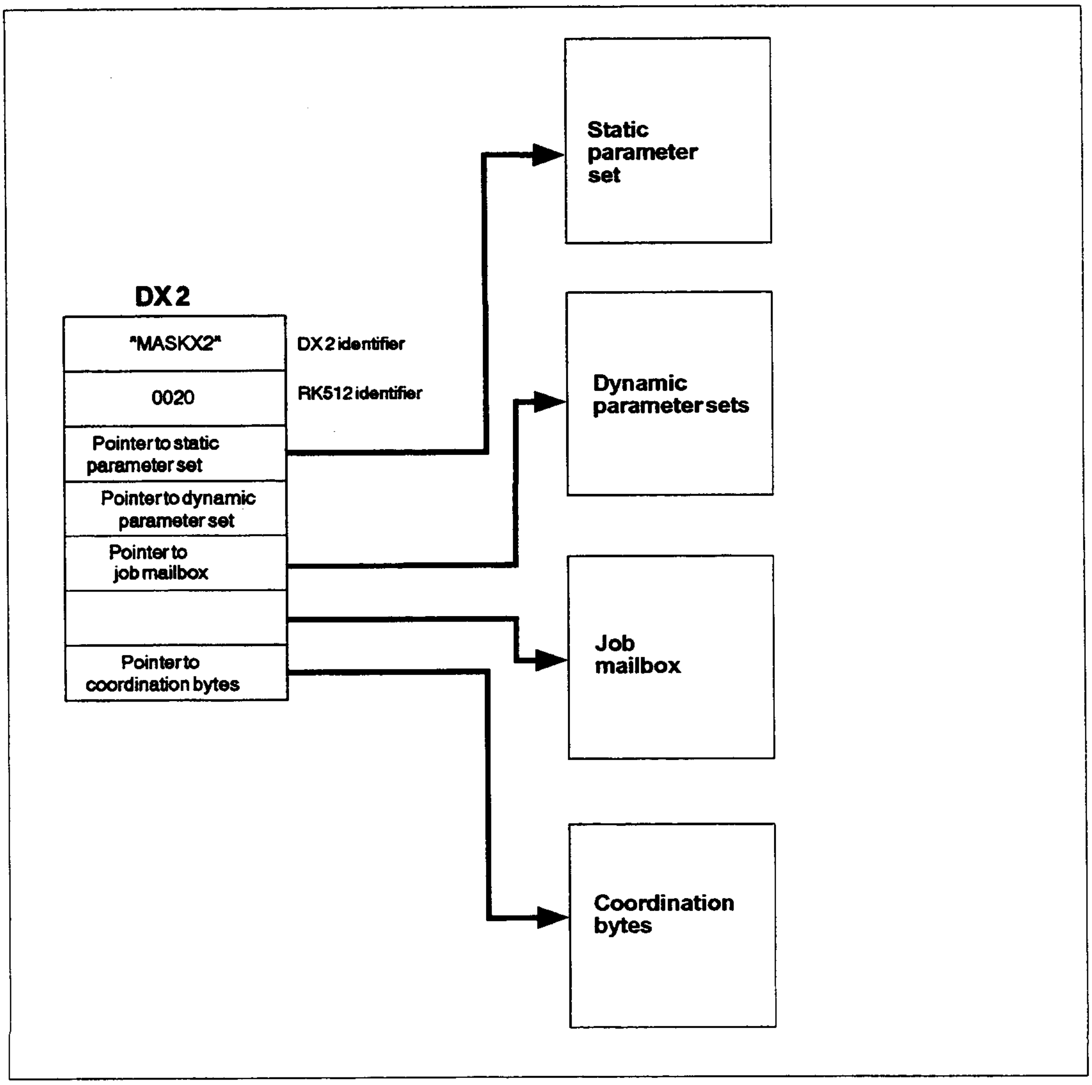


Fig. 4-6 Components of the user program

Operating your CPU 928B and your CPU 948 solely as passive devices

If you only wish to operate your CPU 928B or CPU 948 as a **passive** device, that does not trigger jobs itself, you do not need dynamic parameter sets or a job mailbox for the RK 512 computer link.

Note

If no DX 2 is available in the CPU, initialize the second interface as the PG interface.

Note

You must specify the parameters in DX 2, in the static parameter set and in the dynamic parameter sets before you start up the RK 512 computer link.

With the CPU 928B the parameters in DX 2 are only adopted by the system program following a cold restart.
With the CPU 948 the parameters are also adopted by the system program following power up or an overall reset (loading DX 2 from memory card).

The parameters in the static parameter set (DW 0 to DW 4) are only adopted by the system program following power up or a cold restart.

The parameters in the static parameter set from DW 5 onwards are only adopted by the system program following power up, cold restart and the "reset RK 512 computer link" job.

The parameters in the job mailbox and in the selected dynamic parameter set are adopted by the system program at the beginning of a job. You must only modify these parameters and the send coordination byte (SCB) when no job is currently acitve.

The following chapter provides more detailed information about the values you can use when assigning parameters for RK 512.

4.2 Parameter Assignment

The following sections explain how you can assign parameters to RK 512 and how you should structure the required data blocks.

You have two options:

- either assign parameters directly in STEP 5 by inputting them into data blocks
- or enter the parameters via the menus in the COM PP Parameter Assignment Software and then transfer the data blocks you have created to the CPU using the STEP 5 basic package. Working with COM PP is described in Chapter 3.

4.2.1 Assigning Parameters to DX 2

DX 2 contains the link type. It also contains the pointers to the required parameters and parameter sets. The following table shows the values you can use when assigning parameters to DX 2. The assignment in DX 2 begins at DW 0. All the numerical values are shown in hex.

DW	Parameter	Significance
0	4D41	MA
1	534B	SK
2	5832	X2
Link type		
3	0020	RK 512 computer link
Pointer to static parameter set		
4	44xx or 58xx	DB no. xx or [3] DX no. xx
5	xxxx	from DW no. xxxx
Pointers to dynamic parameter sets [1]		
6	44xx or 58xx	DB no. xx or [3] DX no. xx
7	xxxx	from DW no. xxxx
Pointer to job mailbox [1]		
8	44xx or 58xx	DB no. xx or [3] DX no. xx
9	xxxx	from DW no. xxxx
-		
10 11 12 13	0000 0000 0000 0000	reserved reserved reserved reserved
Pointer to SCB [2]		
14	4D00 or 5300 or 44xx or 58xx	flag or [3] S flag or DB no. xx or DX no. xx
15	xxxx	FW no., SW no. or DW no.

1) These parameters are not necessary if the CPU 928B/CPU 948 is purely passive. If this is the case, enter the value 0000 in these data words.
2) The SCB always occupies one word in memory.
3) Range of values for xx from 3 to FF (3 - 255)

Example

In this example, you can see how to assign the following basic parameters to a DX 2.

- The static parameter set is in data block 10 from data word 4 onwards.
- The dynamic parameter sets are in the extended data block 3 from data word 4 onwards.
- The job mailbox is in data block 6 from data word 2 onwards.
- The SCB is in flag word 6.

With these parameters, DX 2 appears as below:

DX 2

0:	KH = 4D41	'MASKX2'
1:	KH = 534B;	
2:	KH = 5832;	
3:	KH = 0020;	**LINK TYPE** RK 512 COMPUTER LINK
4:	KH = 440A;	**STATIC PARAMETER SET** IN DB 10
5:	KH = 0004;	FROM DATA WORD 4
6:	KH = 5803;	**DYNAMIC PARAMETER SETS** IN DX 3
7:	KH = 0004;	FROM DATA WORD 4
8:	KH = 4406;	**JOB MAILBOX** IN DATA BLOCK 6
9:	KH = 0002;	FROM DATA WORD 2
10:	KH = 0000;	reserved
11:	KH = 0000;	
12:	KH = 0000;	
13:	KH = 0000;	
14:	KH = 4D00;	**SCB** IN FLAG
15:	KH = 0006;	FLAG WORD 6 (see Section 4.2.5)
16:	KH = xxxx;	no longer belongs to the parameter assignment for DX 2, content irrelevant, does not need to exist.
:		
:		

4.2.2 Static Parameter Set

In the static parameter set, you must enter the parameters for the physical and data link layers.

The following table shows the values that you can enter in a data block or extended data block for the static parameter set. The parameter assignment begins at the DW specified in DX 2.

All numerical values are shown in hexadecimal format.

DW (rel.)	Range of values	Significance
Transmission speed		
0	0001	110 bps
	0002	150 bps
	0003	300 bps
	0004	600 bps
	0005	1200 bps
	0006	2400 bps
	0007	4800 bps
	0008	9600 bps
	0009	19200 bps (only with the V.24 submodule and RS 422-A/485 submodule)
Parity		
1	0000	no
	0001	odd
	0002	even
Bits per character		
2	0008	8 bits per character
Stop bits		
3	0001	1 stop bit
	0002	1 + 1/2 stop bits
	0003	2 stop bits
-		
4	0000	reserved
Mode (see note)		
5	0001	RK 512 with 3964 procedure with default values
	0002	RK 512 with 3964R procedure with default values
	0003	RK 512 with 3964 procedure selectable values
	0004	RK 512 with 3964R procedure selectable values
Priority		
6	0000	low
	0001	high

DW (rel.)	Range of values	Significance
Character delay time (see note)		
7	0001 to FFFF	monitoring time 0.01 sec to 655.35 sec (unit: 10 ms)
Acknowledgement delay time (see note)		
8	0001 to FFFF	monitoring time 0.01 sec to 655.35 sec (unit: 10 ms)
Number of connection attempts (see note)		
9	0001 to 00FF	number
Number of repetitions (see note)		
10	0001 to 00FF	number

Note

Data words 7 to 10 are only evaluated if you enter 0003 or 0004 as the mode.

Caution

If you enter 0001 or 0002 as the mode, the default values are used. Remember, however, that data words 7 to 10 must nevertheless exist.

The default values for mode 0001 and mode 0002 are as follows:

	Mode 0001	Mode 0002
Character delay time	220 ms	220 ms
Acknowl. delay time	550 ms	2000 ms
Connection attempts	6	6
Repetitions	6	6

Meaning of the parameters

Transmission speed

Specified in bps

Parity

Number of bits with the value "1" in a string of information bits with a uniform length (e.g. 1 byte). Depending on the value, the parity is "even" or "odd". To check the parity, the string of information bits is extended by a further bit, the parity bit ("0" or "1") which when added to the information bits produces a selectable parity state.

Bits per character

Number of data bits used to form a character (**fixed value of 8**).

Stop bits

Duration of the stop bits, relative to the time required to transmit an information bit. The stop bits follow each character transmitted in a start-stop transmission.

Mode

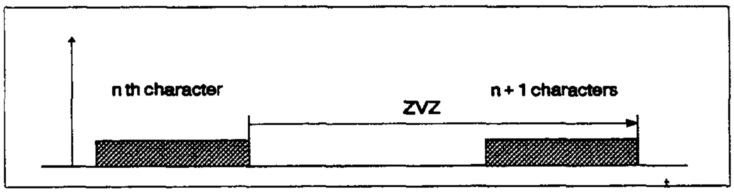
Version of the two possible transmission procedures, 3964 or 3964R, either with default values or with selected values.

Priority

A partner has high priority when its transmit request has priority over the transmit request of another partner. A partner has low priority when its transmit request must give way to the transmit request of the other partner. When using the computer link, the communications partners must have different priority, i.e. A has priority = 1 (high), B must have priority = 0 (low).

Character delay time (ZVZ)

The maximum permitted time interval between two received characters (see following table).



Acknowledgement delay time (QVZ)

The maximum permitted time before receiving the acknowledgement from the partner during connection establishment/termination (see following table).

Transmission speed	Min. permitted ZVZ	Min. permitted QVZ
110 bps	120 ms	120 ms
150 bps	100 ms	100 ms
300 bps	60 ms	60 ms
600 bps	40 ms	40 ms
1200 bps	30 ms	30 ms
2400 bps	20 ms	20 ms
4800 bps	20 ms	20 ms
9600 bps	20 ms	20 ms
19200 bps	20 ms	20 ms

Connection attempts

Maximum number of attempts made by the CPU 928B or the CPU 948 to establish a connection.

Repetitions

Maximum number of message repetitions (including the first message) if errors occur.

Example

In this example, you can see how to assign values for a static parameter set in DB 10 from DW 4 onwards.

DB 10

0:	KH = xxxx;	does not belong to the static parameter set, content
1:	KH = xxxx;	irrelevant
2:	KH = xxxx;	
3:	KH = xxxx;	
4:	KH = 0008;	**TRANSMISSION SPEED** = 9600 bps
5:	KH = 0002;	**PARITY** = 2 = EVEN
6:	KH = 0008;	**BITS PER CHARACTER** = 8
7:	KH = 0001;	**STOP BITS** = 1
8:	KH = 0000;	reserved
9:	KH = 0003;	**MODE** = 3 = 3964 procedure (selectable values)
10:	KH = 0001;	**PRIORITY** = 1 = HIGH
11:	KH = 00FF;	**CHAR. DELAY TIME** = 255 x 0.01 sec (=2.55 sec)
12:	KH = 010A;	**ACK. DELAY TIME** = 266 x 0.01 sec (=2.66 sec)
13:	KH = 0003;	**CONNECTION ATTEMPTS** = 3
14:	KH = 0005;	**REPETITONS** = 5
15:	KH = xxxx;	no longer belongs to the static parameter set,
:		content irrelevant, does not need to exist
:		

In the RK 512 computer link, the parameters for the transmission speed must be the same in the partner and in the CPU 928B/CPU 948.

The parameters for the data link layer in the partner must be set as follows:

- use of the same procedure as in the CPU 928B or in the CPU 948
- opposite priority from that of the CPU 928B or the CPU 948

4.2.3 Dynamic Parameter Sets

The dynamic parameter sets contain the job parameters, relevant to the **partner.**

A dynamic parameter set is **three data words** long and is located in a data block. A data block can contain a maximum of **255 dynamic parameter sets. Each dynamic parameter set has a job number assigned to it. The first dynamic parameter set (relative DW 0 to DW 2) has job number 1, the second (relative DW 3 to DW 5) the job number 2 etc.**

In DX 2, you enter the location of the first dynamic parameter set (job no.1). Before you trigger a SEND or FETCH job, you must have assigned values to the corresponding dynamic parameter set. When you specify a job number in the job mailbox, the selected dynamic parameter set is then used for the SEND or FETCH job.

The following table shows the values that you can enter in the dynamic parameter sets in a data block or extended data block. Parameter assignment begins at the DW specified in DX 2.

All the numerical values are shown in hexadecimal format.

Job no.	DW (rel.)	Parameter	Significance
	SEND/FETCH job with job number 1		
1	Job type		
	0	4144 or 4158 or 4544 or 4558	SEND job with data destination DB or SEND job with data destination DX or FETCH job with data source DB or FETCH job with data source DX
	Data destination (SEND) / data source (FETCH)		
	1	xxyy	DB/DX no. xx from DW no. yy (permitted range see below)
	Coordination flag (CF) (see Section 4.3)		
	2	xxyz	byte no. xx of the coordination flag [1)] range of values 00 to FF (0 to 255) (see note) no. z of the coordination flag range of values 0 to 7 or F CPU number y range of values 1 to 4 (or 0,F) [2)]
	SEND/FETCH job with job number 2		
2	3		
	4		
	5		
.	.	.	.
.	.	.	.
.	.	.	.
	SEND/FETCH job with job number 255		
255	762		
	763		
	764		

1) *If no* coordination flag is specified, you must enter FF as byte no. and F as bit no.

2) The number of the CPU in the partner is only relevant if you are exchanging data via a CP 544 or CP 524/CP 525 with a multiprocessor PLC (S5 135U, S5 155U) with several CPUs.
If you do not specify a CPU, but rather a coordination flag, this field contains the value 0H.
If you do not specify a CPU nor a coordination flag, this field contains the value FH.
In both cases, all the CPUs are possible in the partner.

Note

When you select a coordination flag, make sure that the communications partner can also evaluate this flag. If, for example, a CP 524, CP 525 or CP 544 is the communications partner, the byte numbers of the coordination flag can only have values from 1 to 223.

The following table shows the data blocks and data words permitted as the data destination with a SEND job.

SEND job		**Limit values for assigning parameters to a dynamic parameter set**			
Data source (CPU 928B/ CPU 948)	Data destination (communications partner)	Job type	Data destination		CF permitted
			DB no. DX no.	from DW no.	
DB	DB	AD	3 - 255	0 - 255	yes
DX	DB	AD	3 - 255	0 - 255	yes
DB	DX	AX	3 - 255	0 - 255	yes
DX	DX	AX	3 - 255	0 - 255	yes

The following table shows the data blocks and data words permitted as the data source with a FETCH job.

FETCH job		**Limit values for assigning parameters to a dynamic parameter set**			
Data destination (CPU 928B/ CPU 948)	Data source (communications partner)	Job type	Data source		CF permitted
			DB no. DX no.	from DW no.	
DB	DB	ED	3 - 255	0 - 255	yes
DB	DX	EX	3 - 255	0 - 255	yes
DX	DB	ED	3 - 255	0 - 255	yes
DX	DX	EX	3 - 255	0 - 255	yes

Example

In this example, you can see how to assign parameters for the dynamic parameter sets with job numbers 2 and 3. The dynamic parameter sets are in the extended data block DX 3 from DW 4 onwards.

DX 3

0:	KH = xxxx;	does not belong to the dynamic parameter sets,
1:	KH = xxxx;	content irrelevant
2:	KH = xxxx;	
3:	KH = xxxx;	
4:	KH = yyyy;	**Job with job number 1**
5:	KH = yyyy;	content irrelevant, if job number 1 is not required
6:	KH = yyyy;	
7:	KH = 4144;	**Job with job number 2**
8:	KH = 0B05;	SEND job with data destination DB 11 from DW 5 in CPU 1
9:	KH = 6412;	with CF 100.2
10:	KH = 4558;	**Job with job number 3**
11:	KH =110D;	FETCH job with data source DX 17 from DW 13
12:	KH = FFFF;	onwards from a CPU of the comm. partner without CF
13:	KH = yyyy;	**Job with job number 4**
14:	KH = yyyy;	content irrelevant, does not need to exist if no jobs with job
:		numbers ≥ 4 are required
:		

4

4.2.4 Job Mailbox

The following table shows the values you can enter in the job mailbox in a data block or extended data block. You can only modify the parameters in the job mailbox when no job is currently active. Parameter assignment begins at the DW specified in DX 2.

All numerical values are in hexadecimal format

DW (rel.)	Parameter	Significance
SEND/FETCH job		
0	41xx or 45xx	A - SEND job with job number xx (1 to 255) or E - FETCH job with job number xx (1 to 255)
Condition codeword (CC)[1]		
1	4D00 or 5300 or 44xx or 58xx	in the flag area or in the S flag area or in the data block DB no. xx or in the extended data block DX no. xx
2	xxxx	FW no., SW no. or DW no.
Data source/data destination [2]		
3	44xx or 58xx	in data block DB no. xx or in extended data block DX no. xx
4	xxxx	from DW no. xxxx
5	xxxx	length of the data to be transmitted in words
Parameter error assignment byte (PAFE byte)[1]		
6	4D00 or 5300 or 44xx or 58xx	in the flag area or in the S flag area or in the data block DB no. xx or in the extended data block DX no. xx
7	xxxx	FY no., SY no. or DR no.

1) For detailed information about the condition codeword (CC) and the parameter assignment error byte (PAFE byte) refer to Section 4.5.5.
2) The permitted values are shown in the following tables.

The following table shows the data blocks, data words and length specifications for the data source permitted for a SEND job.

SEND job		**Limit values for assigning parameters to the job mailbox**			
Data source (CPU 928B/ CPU 948)	Data destination (communications partner)	Data source	Data source		Length (words)
			DB no. DX no.	from DW no.	
DB	DB	DB	3 - 255	0 - 2047	1 - 2048
DX	DB	DX	3 - 255	0 - 2047	1 - 2048
DB	DX	DB	3 - 255	0 - 2047	1 - 2048
DX	DX	DX	3 - 255	0 - 2047	1 - 2048

4

The next table shows the data blocks, data words and length specifications for the data destination permitted for a FETCH job.

FETCH job		**Limit values for assigning parameters to the job mailbox**			
Data destination (CPU 928B/ CPU 948)	Data source (communications partner)	Data destination	Data destination		Length (words)
			DB no. DX no.	DW no.	
DB	DB	DB	3 - 255	0 - 2047	1 - 2048
DB	DX	DB	3 - 255	0 - 2047	1 - 2048
DX	DB	DX	3 - 255	0 - 2047	1 - 2048
DX	DX	DX	3 - 255	0 - 2047	1 - 2048

Example

The example shows how to assign parameters for a job mailbox in data block 6 from data word 2 onwards.

DB 6

0:	KH = xxxx;	does not belong to the job mailbox, content irrelevant
1:	KH = xxxx;	
2:	KH = 4503;	**FETCH JOB,** JOB NUMBER 3
3:	KH = 4D00;	**CONDITION CODEWORD** IN FLAG
4:	KH = 0008;	FLAG WORD 8
5:	KH = 5813;	**DATA DESTINATION** IN DX 19
6:	KH = 0014;	FROM DATA WORD 20
7:	KH = 0032;	LENGTH = 50 WORDS
8:	KH = 5300;	**PAFE BYTE** IN S FLAG
9:	KH = 0013;	S FLAG BYTE 19
10:	KH = xxxx;	no longer belongs to the job mailbox, content irrelevant,
11:	KH = xxxx;	does not need to exist
:		
:		

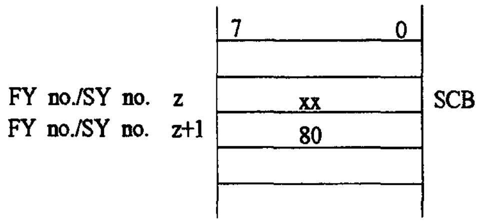
4.2.5 Send Coordination Byte (SCB)

With the SCB you can start the following jobs:

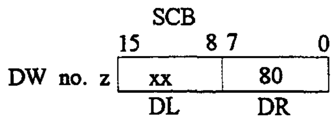
- **SEND jobs**
- **FETCH jobs**
- **reset the** RK 512 computer link

The SCB can be a flag or S flag or can be located in a data block or extended data block. Flags and S flags already exist in the CPU 928B and in the CPU 948, you must set up data blocks and extended data blocks yourself.

Representation in the flag or S flag area



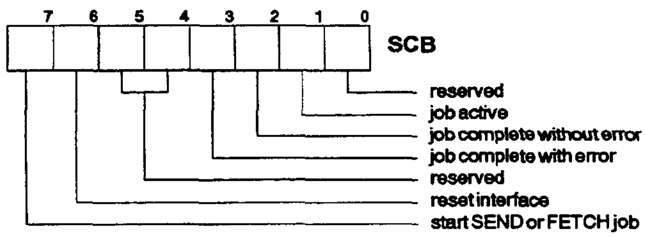
Representation in a data block or extended data block



Note

The system program reserves a data or flag word for the send coordination byte (SCB). The spare byte has 80H written to it following power up or a cold restart.

The send coordination byte (SCB) is structured as follows:



The individual bits or bit groups, providing they are not reserved, are described below.

Bit 1 "Job active"

- set by the system program when a job starts
- reset by the system program when a job is completed
- you can read this bit, to know whether there is any point in starting a new job

Bit 2 "Job complete without error"

- set by the system program when a job is completed without error
- you can read this bit and then reset it, or it is reset by the system program when bit 1 "job active" is set for a new job
- you can read this bit, to know whether a job was completed without error

Bit 3 "Job terminated with error"

- set by the system program when a job is terminated with an error
- you can read this bit and then reset it, or it is reset by the system program when bit 1 "job active" is set for a new job
- you can read this bit, to know whether a job was completed with an error

Bit 6 "Reset interface"

- you can set this bit to reset the interface
- reset by the system program as soon as the system program resets the interface

Bit 7 "Start SEND or FETCH job"

- you can set this bit to start a SEND or FETCH job
- reset by the system program as soon as the system program completes the SEND or FETCH job

When you reset the interface, the parameters of the data link layer in the static parameter set (from DW 5) are read in again:

- mode
- priority
- character delay time
- acknowledgement delay time
- connection attempts
- repetitions

If the SCB bits 6 and 7 are set simultaneously or if SCB bit 6 is set first followed by SCB bit 7 at an interval ≤ 10 ms (CPU 928B) or ≤ 20 ms (CPU 948), the system program first carries out the reset followed by the SEND/FETCH job. If SCB bit 7 is set first followed by SCB bit 6 at an interval ≤ 10 ms (CPU 928B) or ≤ 20 ms (CPU 948), the sequence of the two jobs cannot be guaranteed. By resetting, you abort the currently active partner SEND/FETCH job.

> **Note**
>
> You can only modify the parameters of the physical layer involving the physical transmission characteristics (static parameter sets up to and including DW 4) using a cold restart on the CPU 928B or on the CPU 948. Resetting the interface is not sufficient.

You cannot control jobs on the partner from the user program. The system program processes these jobs automatically in the background. The system program writes the data (for a SEND job on the partner) in the destination DB/DX, or fetches the data (for a FETCH job on the partner) from the source DB/DX. You can only control the data exchange of jobs on the partner using the coordination flags (see Section 4.3).

4.3 Coordination Flags (CF)

Using the coordination flags, you can inhibit or release the overwriting of data resulting from a SEND job on the partner. You can also inhibit or release the transfer of data resulting from a FETCH job on the partner. This allows a coordinated data transmission.

Range of values for the coordination flag:

Byte no.:	0 to 255
Bit no.:	0 to 7

Note

When assigning a coordination flag, check whether the communications partner can also evaluate it. If, e.g. if your partner is a CP 544 or CP 524/525, the byte number of the coordination flag can only have values between 1 and 223.

Each coordination flag occupies a flag bit in the flag area of the CPU 928B or of the CPU 948. You cannot use S flags as coordination flags.

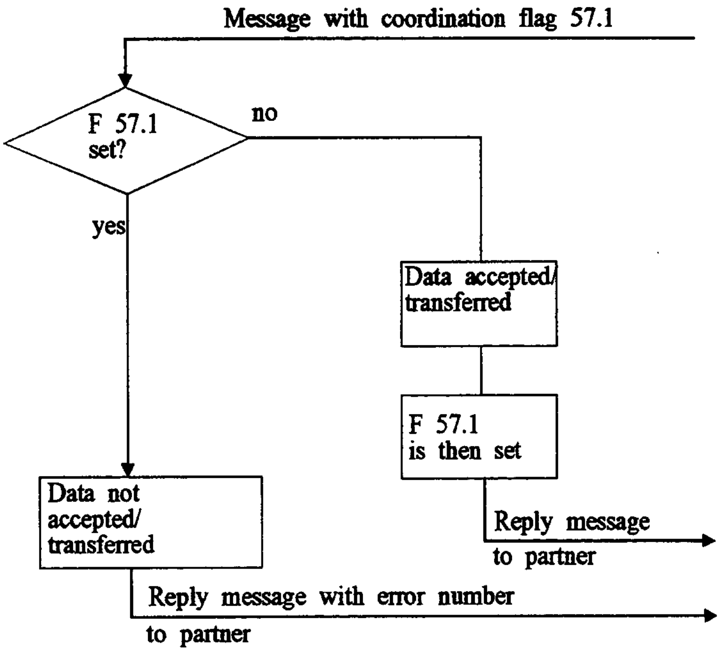
The data are exchanged between the partners in the form of messages (see Section 4.6). If the partner sends a SEND or FETCH message with a coordination flag to the CPU 928B/CPU 948, the CPU 928B/CPU 948 checks whether its coordination flag is set.

If the coordination flag is **not set**, the partner's data is accepted or the requested data is transmitted. Following this, the system program sets the coordination flag.

If the coordination flag is **set**, the CPU 928B/CPU 948 rejects the data exchange with the partner and sends a reply message with an error number to the partner.

Note

After evaluating the data, you must release the data area by resetting the coordination flag in the user program.



Caution

Special situation with FETCH:
After the system sets the coordination flag, the data must still be transmitted by the communications processor. The setting of the coordination flag does not guarantee that the communications partner has received the data free of errors.

4.4 Getting Started

Preparations

You have already installed the interface submodule in the CPU 928B or in the CPU 948 (see Chapter 2), the CPU is plugged into the PLC rack and the power supply is off.

Procedure

We recommend the following procedure for starting the RK 512 computer link:

Step	Action
1	Connect the CPU 928B/CPU 948 to the partner (with a standard connecting cable).
2	Switch on the power supply.
3	Transfer your user program to the CPU 928B. The user program must contain the following **data blocks** or **extended data blocks** to be able to set the SI 2 interface for RK 512: DX 2 DB/DX with the static parameter set DB/DX with the dynamic parameter sets [1)] DB/DX with the job mailbox [1)] DB/DX with the SCB (only if the SCB is located in a DB/DX) [2)]
4	Perform a **cold restart**.
5	Prepare the partner for the RK 512 computer link.

1) Only when the CPU 928B/CPU 948 triggers active jobs itself
2) If the SCB is a flag or S flag word, it already exists in the CPU 928B/CPU 948

CPU 928B

During a cold restart or following power up the system checks that the following exist:

- DX 2,
- the static parameter set
- SCB

and the correct parameter assignment in DX 2

DX 2 exists and has correct parameters

- The system program initializes the SI 2 interface for the RK 512 computer link.
- The red LED SI 2 goes off as soon as the RK 512 computer link is ready for operation.
- The system program resets the SCB bit 7 (i.e. no SEND or FETCH job active).

The RK 512 computer link is ready for operation

DX 2 does not exist or does not have correct parameters

- The CPU remains in the "STOP" mode.
- The red LED SI 2 is lit in addition to the STOP LED.
- The system program outputs an error message to RS 3 and RS 4.

CPU 948

During a cold restart or following power up the system checks that the following exist:

- DX 2,
- the static parameter set
- SCB

and the correct parameter assignment in DX 2

DX 2 exists and has correct parameters

- The system program initializes the SI 2 interface for the RK 512 computer link.
- The red LED SI 2 goes off as soon as the RK 512 computer link is ready for operation.
- The system program resets the SCB bit 7 (i.e. no SEND or FETCH job active).

The RK 512 computer link is ready for operation

DX 2 does not exist or does not have correct parameters

- The system program does not initialize the SI 2 interface for the RK 512 computer link
- The red LED SI 2 does not go off

4.5 Operation

Preparations

The SI 2 interface has parameters assigned for the RK 512 computer link. The SI 2 LED is off. RK 512 is ready for operation.

Method

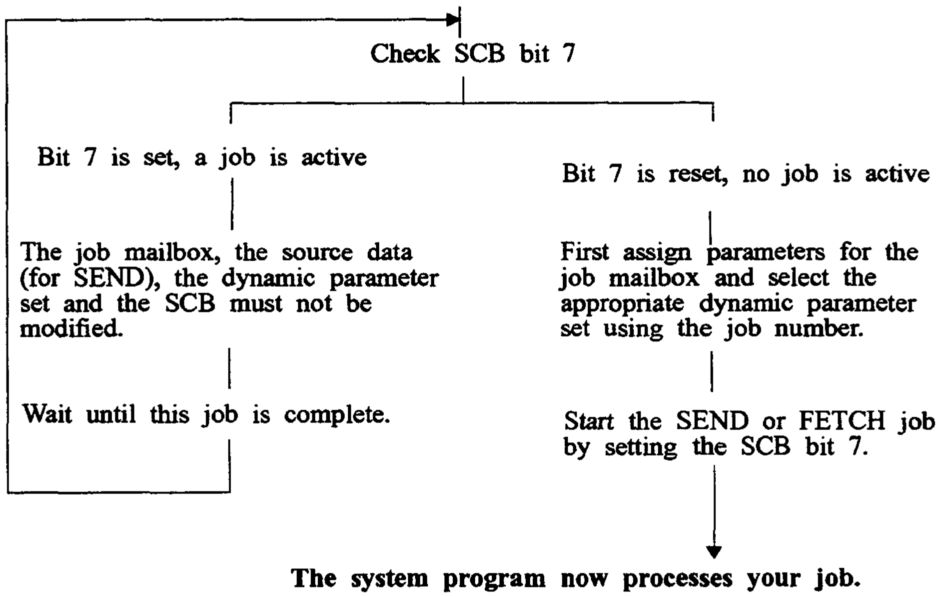
The system program now checks constantly whether the send coordination byte (SCB) exists. If it does not exist (e.g. because the DB/DX has been deleted) the system program calls OB 35. The error number is entered in ACCU 1 (see Chapter 11).

The system program now processes the jobs you have programmed in your user program, or the jobs triggered by the communications partner. The following sections explain how to program the jobs in your user program.

4.5.1 SEND/FETCH Job

If you want to program a SEND/FETCH job, remember the following sequence:

Start job:



The following diagram illustrates the reactions that can occur if the SCB bit 7 is set and the system program is processing your job.

SCB bit 7 is set

The job mailbox and the dynamic parameter set exist and the parameter assignment is correct and within the job limits.

The job is processed. The system program begins with data transmission or requests data.

The job was completed withour errors.

The system program

- transferred the data for a SEND job from the source block to the destination block of the partner or transferred the data for a FETCH job from the source block of the partner to the destination block
- reset the SCB bit 7.

The job was terminated with an error.

The system program

- called OB 35: ACCU 1 contains the error number
- set the bit "job terminated with error" in the CC and entered an error number.
- reset the SCB bit 7.

The job mailbox or the dynamic parameter set does not exist or the parameter assignments are incorrect or outside the job limits.

The system program

- did not process the job
- entered an error message in the parameter assignment byte (PAFE) assuming the PAFE byte has been correctly assigned and a parameter assignment error has occurred in connection with the job mailbox
- entered an error message and set the "job complete with error" bit in the CC assuming the CC is correct
- called OB 35 and entered an error number in ACCU 1
- reset the SCB bit 7.

Note

You can only start a new job when the previous job is completed.

4.5.2 Resetting the Interface

You have set the SCB bit 6 and therefore started a reset job. The following diagram shows the possible reactions.

SCB bit 6 is set

The CPU 928B/CPU 948 is currently processing a job from the partner

- the partner's job is aborted with an error
- OB 35 has been called: ACCU 1 contains the error number for the aborted job of the partner.
- the static parameters form DW 5 onwards are read and evaluated
- when the reset job is complete, the SCB bit 6 is reset

The CPU 928B/CPU 948 is not processing a job from the partner

- the static parameters from DW 5 onwards are read in and evaluated
- when the reset job is complete, the SCB bit 6 is reset.

4.5.3 Job from the Partner

Jobs from the partner are not processed by the user program, but by the system program automatically in the background. To make sure that the system program only works with consistent data, you can use coordination flags in your user program to prevent data being overwritten (partner SEND) or read (partner FETCH) (see Section 4.3 "Coordination Flags (CF)").

The diagram on the following page shows the situations that can occur following a job triggered by the partner.

CPU 928B

Job from partner active

- The CPU 928B is not in the STOP mode or wait state (program test)
  - The partner job is completed without error
    - The system program
      - transferred the data from the partner to the dest. block for a partner SEND job
      - fetched the data from the source block and sent them to the partner for a partner FETCH job.
  - The partner job is terminated with an error
    - The system program
      - aborted the partner job with an error
      - called OB 35: ACCU 1 contains the error number
      - sent a reply message with an error number to the partner.
- The CPU 928B is in the STOP mode or wait state (program test)
  - The system program
    - aborted the partner job with an error
    - sent a reply message with the error number to the partner
    - sent a reply message with the error number to the partner. called OB 35 after leaving the STOP or wait state ACCU 1 contains the error number

CPU 948

Job from partner active

- The partner job is completed without error
  - The system program
    - transferred the data from the partner to the dest. block for a partner SEND job
    - fetched the data from the source block and sent them to the partner for a partner FETCH job.
- The partner job is terminated with an error
  - The system program
    - aborted the partner job with an error
    - called OB 35: ACCU 1 contains the error number
    - sent a reply message with an error number to the partner.

4.5.4 Response in Special Operating Modes

Stop mode
CPU 928B

In the "STOP" mode, user programs are not processed, i.e. you cannot start a new job. When the CPU changes from "RUN" to "STOP", the following applies to an already active job started by the user program:

- The job is stopped with an error, if there is still data to be transferred between the operating system processor and the dual-port RAM.
- The job is completed normally if there is no more data to be transferred between the operating system processor and the dual-port RAM.

If the partner triggers a job in the "STOP" mode, this is acknowledged negatively. An appropriate error message is entered in the error area of the interface (refer to Chapter 11 "Error Messages"). When the CPU changes from the "RUN" mode to the "STOP" mode, an already active job started by the partner is stopped with an error, i.e. negatively acknowledged. A corresponding error message is entered in the error area of the interface (see Chapter 11 "Error Messages")

CPU 948

In contrast to the CPU 928B, the CPU 948 also supports communication in "soft" stop and in the wait mode (program test).

Wait mode
CPU 928

The CPU 928B goes into the wait mode when the PG online function "PROGRAM TEST" is active. In the wait mode user programs are not processed, i.e. you cannot start a new job. An already active job started by the user is continued and completed after the wait mode is exited.

If the partner triggers a job in the wait mode, this is acknowledged negatively. An appropriate error message is entered in the error area of the interface (refer to Chapter 11 "Error Messages"). When the CPU changes to the wait mode, an already active job started by the partner is negatively acknowledged. A corresponding error message is entered in the error area of the interface (see Chapter 11 "Error Messages").

CPU 948

In contrast to the CPU 928B, the CPU 948 also supports communication in "soft" stop and in the wait mode (program test).

4.5.5 Error Handling

If an error occurs, this is indicated by an error number:

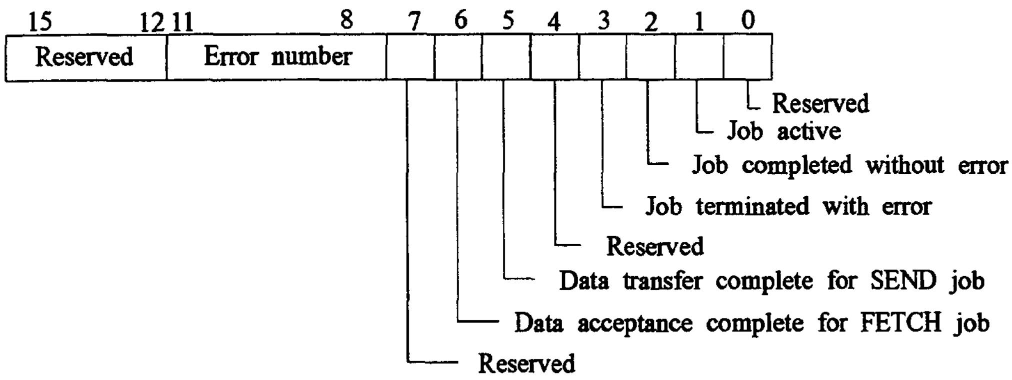
- in the CC condition codeword
- in the PAFE byte
- in ACCU 1 with an error number when OB 35 is called

Errors occurring when a job from the partner is processed (partner SEND/partner FETCH) are signalled to the partner with an error number in the reply message REPMES (see Section 4.6.3).

Condition codeword (CC)

You must specify the condition codeword in the job mailbox. It can be a flag, S flag or data word in a data block or extended data block. Flags are S flags always exist in the CPU 928B; you must first create data blocks and extended data blocks.

The system program enters information about the **status of job processing** in the condition codeword (e.g. "job active" or "job completed without error or job terminated with error"). If the job was terminated with an error, bits 8 to 11 also include an error number (see Section 11.2.2). With the aid of this error number, you can localize and remedy the error more quickly.



The meaning of the individual bits or groups of bits (except for the reserved bits) is described below.

Bit 1 "Job active"

- Is set by the system program when a job is started (SCB bit 7 set); this is only possible if the job mailbox has a correct parameter assignment.
- Is reset by the system program when the job is complete.

You can evaluate this bit if you want to know whether or not you can start a new job.

Bit 2 "Job completed without error"

- Set by the system program when the job is completed without errors.
- You can evaluate and then reset this bit or it is reset by the system program as soon as bit 1 "job active" is set for a new job with the same CC.
- You can evaluate this bit if you want to know whether or not the job was completed without errors.

Bit 3 "Job terminated with error"

- Set by the system program when the job was terminated with an error. In this case, an **error number** is entered in bits 8 to 11.
- After evaluation, you can reset this bit or it is reset by the system program as soon as bit 1 "Job active" is set for a new job with the same CC.
- You can evaluate this bit if you want to know whether or not a job was completed with errors. From bits 8 to 11 of the condition codeword, you can read the error number that describes the cause of the error. All the error numbers - descriptions and remedies - are listed in Section 11.2.

Bit 5 "Data transfer complete for SEND job"

- This is set by the system program when the data for a SEND job have been fetched completely from the data source.
- After evaluation, you can reset this bit or it is reset by the system program as soon as data are fetched from the data source for a new job to which this condition codeword is assigned.
- You can evaluate this bit if you want to know whether the data have already been completely fetched from the data source. Following this, you can once again write to the data source even though the SEND job is still active.

Bit 6 "Data acceptance completed for FETCH job"

- Set by the system program when the data for a FETCH job have been transferred completely to the data destination.
- After evaluation, you can reset this bit or it is reset by the system program as soon as data are transferred to the data destination for a new job that has this condition codeword assigned to it.
- You can evaluate this bit if you want to know whether or not the data have already been transferred completely to the data destination. Following this, you can evaluate the requested data although the FETCH job is still active.

Bits 8 to 11 "Error number"

- If bit 3 = 1 ("job terminated with error") an error number is also entered in these bits. All the error numbers, a description and the remedies are listed in Section 11.2.

Parameter assignment error byte (PAFE Byte)

You must specify the parameter assignment error byte in the job mailbox. This can be a flag byte, S flag byte or data word (right-hand data byte) in a data block or extended data block. Flags and S flags already exist in the CPU 928B and in the CPU 948; you must create data blocks and extended data blocks yourself.

The PAFE byte indicates errors associated directly with the **parameter assignment in the job mailbox**

7 ... 4	3 ... 1	0
Error number	Reserved	

0 = no error
1 = error

If bit 0 = 1 in the PAFE byte, a hexadecimal error number is entered in bits 4 to 7 (see Section 11.2.4).

4.5.6 Multiprocessor Mode

If you want to exchange data with more than one CPU in an S5-135U or S5-155U (via a CP 544 or CP 524/525), you must specify a CPU number in the dynamic parameter set. For more detailed information about assigning parameters to the dynamic parameter set, refer to Section 4.2.3.

If your partner is a CPU 928B or a CPU 948, you do not need a CPU number. If the partner sends a CPU number, this is not evaluated by the CPU.

4.6 Data Exchange

The data exchange between the partners takes place at three levels.

- The hardware level (physical layer)
- The procedure level (data link layer)
- The message level (transport layer)

4.6.1 Hardware Level

The hardware level covers the physical conditions for the data transmission. You can plug in various interface submodules in the CPU 928B/CPU 948 (TTY, V.24 or RS422-A/485 submodules). For more detailed information about the interface submodules, refer to the S5-135U/155U System Manual.

4.6.2 Procedure Level

All the messages of the RK 512 (command and reply messages) are transmitted by the procedure level of the computer link, i.e. transmitted or received using the 3964 or 3964R procedure.
The procedure provides the messages with a start or end sequence, supplies the blocks or data with a block check character (only with the 3964R procedure) and automatically initiates repetitions of messages if the receiver sends a negative acknowledgement.

4.6.3 Message Level

The message level corresponds to the transport layer of the ISO/OSI model.
The CPU 928B/CPU 948 transmits a command message to the partner. This then sends a reply message to the CPU.

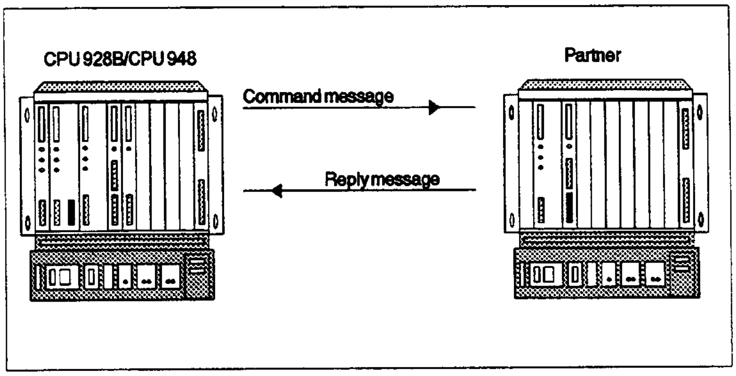


Fig. 4-7 Command and reply messages

Command messages are either SEND messages (for a SEND job) or FETCH messages (for a FETCH job).

The SEND message consists of a message header and data. In response to a SEND message, the partner sends a reply message without data.

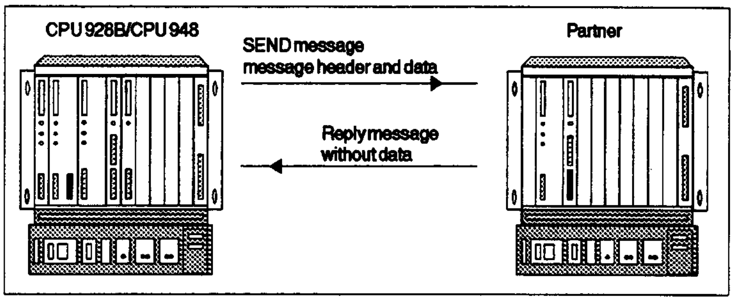


Fig. 4-8 SEND job

The FETCH message consists only of the message header. In response to a FETCH message, the partner sends a reply message with data.

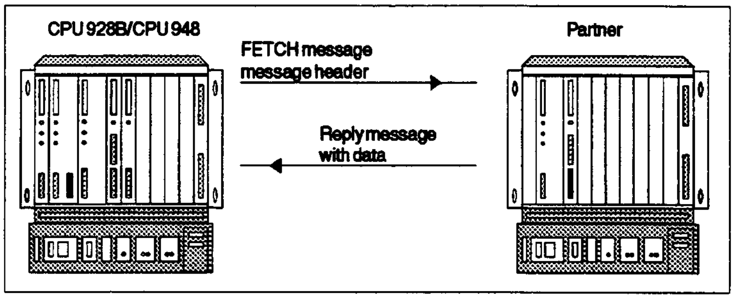


Fig. 4-9 FETCH job

If the volume of data exceeds 128 bytes, follow-on messages are sent for all message types.

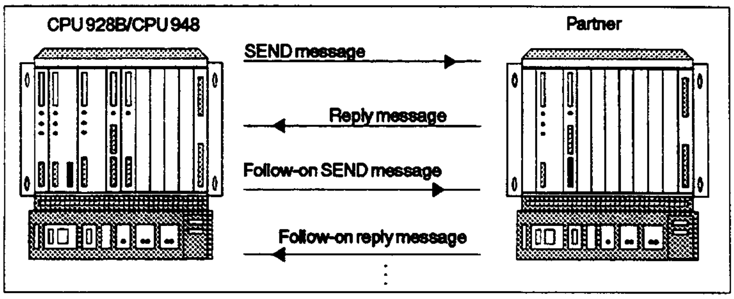


Fig. 4-10 Messages and follow-on messages

Structure of command messages

In RK 512, there are two types of command messages, SEND messages and FETCH messages.

The SEND message consists of a **message header and data**, the FETCH message only consists of the **message header**.

The **message header** consists of **10 bytes**. For a SEND message, it contains information about the data destination, for a fetch message, it contains information about the data source.

Structure of the message header

All the values are in hexadecimal format.

Byte	Content	Meaning	
1	00 or FF	in command messages in follow-on command messages	
2	00		
3	41 4F 45	'A' - for SEND job with destination DB or 'O' - for SEND job with destination DX or 'E' - for FETCH job	
4	44 58	data to be transmitted from a DB data to be transmitted from a DX	
5	xx	DB/DX no.	**data destination** for SEND job, **data source** for FETCH job
6	xx	from DW no. onwards	
7	xx	Length high byte	**length** of the data to be transmitted in words
8	xx	Length low byte	
9	xx	**Coordination flag** - byte number if you have not specified a coordination flag, the value FF_H is entered here	
10	xx	Bits 0 to 3:	**coordination flags** - bit number if you have not specified a coordination flag, the value F_H is entered here
		Bits 4 to 7:	**CPU number** of the partner (number between 1 and 4) if you do not specify a CPU number but a CF, the value 0_H is entered here if you do not specify a CPU number nor a CF, the value F_H is entered here

Note

The message header of a follow-on SEND or follow-on FETCH message only consists of bytes 1 to 4.

Once a command message has been transmitted, RK 512 waits for a reply message from the partner within a monitoring time. The monitoring time depends on the transmission speed (bps):

from	1200 to 19 200 bps	5 sec
	600 bps	7 sec
	300 bps	10 sec
	150 bps	15 sec
	110 bps	20 sec

You specify the transmission speed in the static parameter set.

Structure of the reply message

The reply message without data consists of **4 bytes** and contains information about the processing of the job.

All values are in hexadecimal format.

Byte	Content	Meaning
1	00 or FF	in reply messages in follow-on reply messages
2	00	
3	00	
4	00 or xx	 error number from partner, see Chapter 11

The reply message with data (for a FETCH job) also contains the data from byte 5 onwards.

4.7 Description of the Messages

4.7.1 Sequence of a SEND Job in Detail

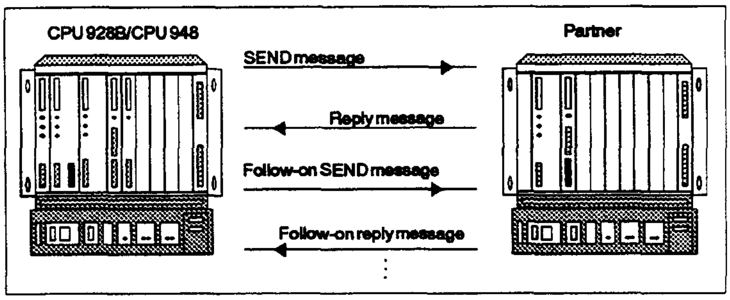


Fig. 4-11 SEND job

The SEND messages and reply messages are sent in the order shown below:

Active partner:	sends a SEND message. This contains the message header and data.
Passive partner:	receives the message, checks the message header and the data and acknowledges with a reply message.
Active partner:	receives the reply message. If the volume of user data exceeds **128 bytes** it sends a follow-on SEND message. This contains bytes 1 to 4 of the message header and further data.
Passive partner:	receives the follow-on SEND message, checks the message header and the data and acknowledges with a follow-on reply message.

. .

. .

Follow-on SEND messages and follow-on reply messages are sent when the user data exceeds **128 bytes**. The first byte of the message header in the follow-on SEND or follow-on reply message contains FFH instead of 00H as the identifier.

If the SEND message is not received free of errors, the partner enters an error number in the fourth byte of the reply message.

The following page illustrates the sequence of a SEND message and reply message. The SEND message has a user data length of maximum 128 bytes and is received without errors (the fourth byte in the reply message does not contain an error number).

	CPU 928B/CPU 948		Partner	
Meaning		**Content**		
SEND message				
Start character		STX 02H	→	
Positive ack.	←	DLE 10H		
		1st byte 00H	→	Message header
		2nd byte 00H	→	Message header
SEND job		3rd byte 41H	→	Message header
Data block		4th byte 44H	→	Message header
Data dest. DB 10		5th byte 0AH	→	Message header
DW 01		6th byte 01H	→	Message header
Length		7th byte 00H	→	Message header
50 DW		8th byte 32H	→	Message header
no CF		9th byte FFH	→	Message header
All CPUs		10th byte FFH	→	Message header
1st data byte		11th byte	→	User data
2nd data byte		12th byte	→	User data
:		:		User data
nth data byte		xth byte	→	User data
End identifier		DLE 10H	→	
End identifier		ETX 03H	→	
Positive ack.	←	DLE 10H		
Reply message				
Start character	←	STX 02H		
Positive ack.		DLE 10H	→	
	←	1st byte 00H		
	←	2nd byte 00H		
	←	3rd byte 00H		
Error number	←	4th byte 00H		
End identifier	←	DLE 10H		
End identifier	←	ETX 03H		
Positive ack.		DLE 10H	→	

Fig. 4-12 Sequence of an error-free SEND message with reply message with RK 512 using the 3964 procedure

Block check character

Remember that when using the 3964R procedure, the block check character "BCC" (longitudinal parity) is sent after the end identifiers DLE and ETX. For more detailed information about the 3964 and 3964R procedures, refer to Section 5.5

The following page illustrates the sequence of a follow-on SEND message with a follow-on reply message. The follow-on SEND message has a user data length of maximum 128 bytes and is received free of errors (the fourth byte in the follow-on reply message does not contain an error number).

		CPU 928B/CPU 948		Partner	
Meaning		**Content**			
Folllow-on SEND message					
Start character	→	STX 02H	→		
Positive ack.	←	DLE 10H	←		
Follow-on message	→	1st byte FFH	→		Mess-age header
	→	2nd byte 00H	→		
SEND job	→	3rd byte 41H	→		
Data block	→	4th byte 44H	→		
129th data byte	→	5th byte	→		User data
130th data byte	→	6th byte	→		
:		:			
nth data byte	→	xth byte	→		
End identifier	→	DLE 10H	→		
End identifier	→	ETX 03H	→		
Positive ack.	←	DLE 10H	←		
Follow-on reply message					
Start character	←	STX 02H	←		
Positive ack.	→	DLE 10H	→		
Follow-on reply message	←	1st byte FFH	←		
	←	2nd byte 00H	←		
	←	3rd byte 00H	←		
Error number	←	4th byte 00H	←		
End identifier	←	DLE 10H	←		
End identifier	←	ETX 03H	←		
Positive ack.	→	DLE 10H	→		

Fig. 4-13 *Sequence of a* follow-on SEND message with a follow-on reply message using the RK 512 with the 3964 procedure

4.7.2 Sequence of a FETCH Job in Detail

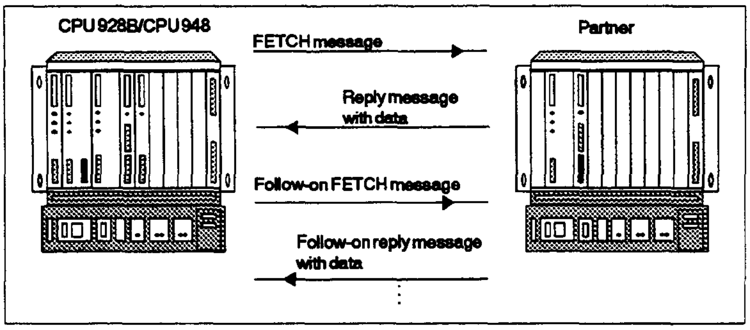


Fig. 4-14 FETCH job

4

FETCH messages and reply messages are exchanged in the following sequence:

Active partner:	sends a FETCH message. This contains the message header.
Passive partner:	receives a message, checks the message header and acknowledges with a reply message. This contains data.
Active partner:	receives the reply message. If the user data exceeds **128 bytes** it sends a follow-on FETCH message. This contains bytes 1 to 4 of the message header.
Passive partner:	receives the follow-on FETCH message, checks the message header and acknowledges with a follow-on reply message. This contains further data.
.	.
.	.

Follow-on FETCH messages are sent when the user data exceeds **128 bytes**. The first byte in the message header of a follow-on FETCH or follow-on REPLY message has the identifier FFH instead of 00H.

If the FETCH message is not received free of errors, the partner enters an error number in the fourth byte of the reply message. The requested data are then **not** transmitted.

The following page illustrates the sequence of a FETCH message with a reply message. The reply message to the FETCH request has a user data length of maximum 128 bytes and is received free of errors (the fourth byte in the reply message does not contain an error number).

Meaning	CPU 928B/CPU 948	Content	Partner	
FETCH message				
Start character	———	STX 02H	———→	
Positive ack.	←———	DLE 10H	———	
	———	1st byte 00H	———→	Message header
	———	2nd byte 00H	———→	
FETCH job	———	3rd byte 45H	———→	
Data block	———	4th byte 44H	———→	
Data source DB 100	———	5th byte 64H	———→	
DW 100	———	6th byte 64H	———→	
Length	———	7th byte 00H	———→	
50 DW	———	8th byte 32H	———→	
CF = F 10.7	———	9th byte 0AH	———→	
only CPU 1	———	10th byte 17H	———→	
End identifier	———	DLE 10H	———→	
End identifier	———	ETX 03H	———→	
Positive ack.	←———	DLE 10H	———	
Reply message with data				
Start character	←———	STX 02H	———	
Positive ack.	———	DLE 10H	———→	
	←———	1st byte 00H	———	
	←———	2nd byte 00H	———	
	←———	3rd byte 00H	———	
Error number	←———	4th byte 00H	———	
1st data byte	←———	5th byte	———	User data
2nd data byte	←———	6th byte	———	
:		:		
nth data byte	←———	xth byte	———	
End identifier	←———	DLE 10H	———	
End identifier	←———	ETX 03H	———	
Positive ack.	———	DLE 10H	———→	

Fig. 4-15 Sequence of an error-free FETCH message with reply message using RK 512 with the 3964 procedure

Block check character

Remember that when using the 3964R procedure, the block check character "BCC" (longitudinal parity) is sent after the end identifiers DLE and ETX. For more detailed information about the 3964 and 3964R procedures, refer to Section 5.5

The following page illustrates the sequence of a follow-on FETCH message with a follow-on reply message. The follow-on reply message for the FETCH request has a user data length of maximum 128 bytes and is received free of errors (the fourth byte in the follow-on reply message does not contain an error number).

Meaning	CPU 928B/CPU 948	Content	Partner	
Follow-on FETCH message				
Start character	→	STX 02H	→	
Positive ack.	←	DLE 10H	←	
Follow-on message	→	1st byte FFH	→	Message header
	→	2nd byte 00H	→	
FETCH job	→	3rd byte 45H	→	
Data block	→	4th byte 44H	→	
End identifier	→	DLE 10H	→	
End identifier	→	ETX 03H	→	
Positive ack.	←	DLE 10H	←	

Meaning	CPU 928B/CPU 948	Content	Partner	
Follow-on reply message				
Start character	←	STX 02H	←	
Positive ack.	→	DLE 10H	→	
Follow-on reply message	←	1st byte FFH	←	
	←	2nd byte 00H	←	
	←	3rd byte 00H	←	
Error number	←	4th byte 00H	←	
129th data byte	←	5th byte	←	User data
130th data byte	←	6th byte	←	
:		:		
nth data byte	←	xth byte	←	
End identifier	←	DLE 10H	←	
End identifier	←	ETX 03H	←	
Positive ack.	→	DLE 10H	→	

Fig. 4-16 Sequence of a follow-on SEND message with a follow-on reply message using the RK 512 with the 3964 procedure

4.7.3 Quasi Duplex Mode

Quasi duplex means: the partners can send messages at any point in time except when the partner is currently transmitting.

If both partners want to transmit simultaneously, a command message might for example arrive before the reply message from the partner.

In the following example, the first follow-on reply message (*) for the first follow-on SEND message is only sent after the **SEND message from the partner**.

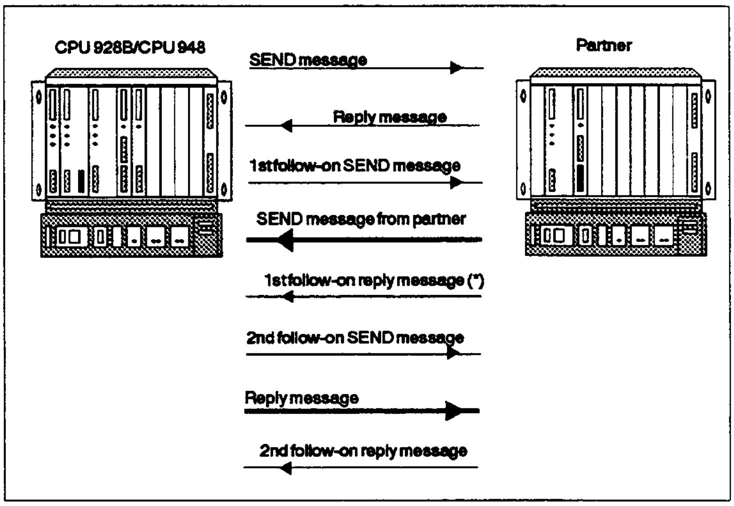


Fig. 4-17 Quasi duplex mode

4.8 Processing Times

This section lists the typical processing times for a SEND or FETCH job.

4.8.1 Processing Times for the CPU 928B

The processing times are subject to the following conditions:

- two CPU 928Bs are connected
- both CPUs have parameters assigned for RK 512 in mode 1 (even parity, 1 stop bit)
- the active CPU has the higher priority

The processing time is the time
- from starting the job on the active CPU 928B (the user program sets SCB bit 7)
- until the job is completed on the active CPU 928B (the system program resets SCB bit 7)

SEND job

The times were measured
- with different transmission speeds and
- with various amounts of user data

No. of DWs	Transmission speed								
	110	150	300	600	1200	2400	4800	9600	19200
1	2.681s	1.946s	0.975s	0.492s	0.259s	0.139s	0.079s	0.049s	0.039s
2	2.865s	2.099s	1.053s	0.529s	0.279s	0.149s	0.089s	0.059s	0.039s
4	3.254s	2.406s	1.203s	0.603s	0.309s	0.169s	0.099s	0.059s	0.039s
8	4.054s	2.978s	1.496s	0.749s	0.389s	0.200s	0.109s	0.069s	0.049s
16	5.755s	4.219s	2.111s	1.059s	0.539s	0.279s	0.149s	0.089s	0.059s
32	8.855s	6.493s	3.247s	1.629s	0.825s	0.419s	0.219s	0.124s	0.079s
64	15.236s	11.193s	5.600s	2.800s	1.409s	0.719s	0.369s	0.199s	0.109s
128	29.918s	21.940s	10.973s	5.493s	2.759s	1.389s	0.709s	0.369s	0.199s
192	44.584s	32.692s	16.354s	8.182s	4.109s	2.069s	1.049s	0.549s	0.294s
256	59.234s	43.440s	21.728s	10.876s	5.457s	2.749s	1.389s	0.719s	0.389s
512	117.872s	86.440s	43.242s	21.645s	10.856s	5.460s	2.769s	1.429s	0.759s
1024	235.144s	172.432s	86.264s	43.186s	21.658s	10.892s	5.509s	2.839s	1.509s
1536	352.416s	258.440s	129.304s	64.752s	32.478s	16.344s	8.274s	4.262s	2.276s
2048	469.696s	344.448s	172.344s	86.300s	43.280s	21.772s	11.015s	5.669s	3.019s

FETCH job

The times were measured

- with different transmission speeds and
- with various amounts of user data

No. of DWs	Transmission speed								
	110	**150**	**300**	**600**	**1200**	**2400**	**4800**	**9600**	**19200**
1	2.653s	1.945s	0.981s	0.491s	0.249s	0.129s	0.069s	0.041s	0.029s
2	2.854s	2.093s	1.049s	0.528s	0.269s	0.139s	0.079s	0.049s	0.029s
4	3.254s	2.387s	1.194s	0.600s	0.303s	0.159s	0.089s	0.049s	0.030s
8	4.054s	2.983s	1.486s	0.746s	0.379s	0.194s	0.100s	0.059s	0.039s
16	5.771s	4.230s	2.111s	1.059s	0.533s	0.269s	0.139s	0.079s	0.049s
32	8.856s	6.493s	3.256s	1.628s	0.819s	0.415s	0.210s	0.119s	0.069s
64	15.259s	11.193s	5.594s	2.800s	1.403s	0.709s	0.359s	0.189s	0.099s
128	29.921s	21.940s	10.975s	5.492s	2.750s	1.381s	0.699s	0.359s	0.190s
192	44.574s	32.688s	16.355s	8.182s	4.099s	2.060s	1.039s	0.539s	0.289s
256	59.240s	43.444s	21.729s	10.873s	5.447s	2.739s	1.379s	0.709s	0.379s
512	117.868s	86.432s	43.244s	21.639s	10.846s	5.454s	2.757s	1.419s	0.759s
1024	235.160s	172.448s	86.264s	43.176s	21.646s	10.883s	5.500s	2.824s	1.500s
1536	352.432s	258.440s	129.292s	64.738s	32.467s	16.332s	8.246s	4.255s	2.256s
2048	469.696s	344.448s	172.336s	86.288s	43.268s	21.761s	11.004s	5.662s	3.014s

4.8.2 Processing Times for the CPU 948

The processing times are subject to the following conditions:

- two CPU 948s are connected
- both CPUs have parameters assigned for RK 512 in mode 1 (even parity, 1 stop bit)
- the active CPU has the higher priority

The processing time is the time
- from starting the job on the active CPU 948 (the user program sets SCB bit 7)
- until the job is completed on the active CPU 948 (the system program resets SCB bit 7)

SEND job

The times were measured
- with different transmission speeds and
- with various amounts of user data

No. of DWs	Transmission speed								
	110	150	300	600	1200	2400	4800	9600	19200
1	2.653s	1.950s	0.975s	0.499s	0.269s	0.149s	0.096s	0.069s	0.049s
2	2.855s	2.106s	1.047s	0.539s	0.289s	0.159s	0.099s	0.069s	0.050s
4	3.265s	2.386s	1.202s	0.609s	0.319s	0.175s	0.109s	0.069s	0.059s
8	4.054s	2.977s	1.489s	0.758s	0.399s	0.219s	0.129s	0.079s	0.059s
16	5.761s	4.225s	2.116s	1.069s	0.549s	0.289s	0.169s	0.099s	0.069s
32	8.859s	6.498s	3.250s	1.639s	0.839s	0.439s	0.239s	0.137s	0.089s
64	15.261s	11.191s	5.597s	2.810s	1.419s	0.729s	0.379s	0.209s	0.129s
128	29.920s	21.944s	10.977s	5.503s	2.769s	1.409s	0.729s	0.389s	0.219s
192	44.582s	32.696s	16.354s	8.197s	4.119s	2.079s	1.069s	0.559s	0.309s
256	59.241s	43.449s	21.734s	10.888s	5.469s	2.757s	1.409s	0.739s	0.399s
512	117.887s	86.457s	43.248s	21.659s	10.869s	5.479s	2.779s	1.441s	0.779s
1024	235.174s	172.475s	86.286s	43.208s	21.680s	10.914s	5.529s	2.859s	1.539s
1536	352.462s	258.495s	129.328s	64.765s	32.489s	16.351s	8.279s	4.279s	2.294s
2048	469.748s	344.512s	172.377s	86.321s	43.300s	21.788s	11.030s	5.699s	3.049s

FETCH job

The times were measured
- with different transmission speeds and
- with various amounts of user data

No. of DWs	Transmission speed								
	110	150	300	600	1200	2400	4800	9600	19200
1	2.653s	1.950s	0.975s	0.499s	0.269s	0.149s	0.096s	0.069s	0.049s
2	2.855s	2.106s	1.047s	0.539s	0.289s	0.159s	0.099s	0.069s	0.050s
4	3.265s	2.386s	1.202s	0.609s	0.319s	0.175s	0.109s	0.069s	0.059s
8	4.054s	2.977s	1.489s	0.758s	0.399s	0.219s	0.129s	0.079s	0.059s
16	5.761s	4.225s	2.116s	1.069s	0.549s	0.289s	0.169s	0.099s	0.069s
32	8.859s	6.498s	3.250s	1.639s	0.839s	0.439s	0.239s	0.137s	0.089s
64	15.261s	11.191s	5.597s	2.810s	1.419s	0.729s	0.379s	0.209s	0.129s
128	29.920s	21.944s	10.977s	5.503s	2.769s	1.409s	0.729s	0.389s	0.219s
192	44.582s	32.696s	16.354s	8.197s	4.119s	2.079s	1.069s	0.559s	0.309s
256	59.241s	43.449s	21.734s	10.888s	5.469s	2.757s	1.409s	0.739s	0.399s
512	117.887s	86.457s	43.248s	21.659s	10.869s	5.479s	2.779s	1.441s	0.779s
1024	235.174s	172.475s	86.286s	43.208s	21.680s	10.914s	5.529s	2.859s	1.539s
1536	352.462s	258.495s	129.328s	64.765s	32.489s	16.351s	8.279s	4.279s	2.294s
2048	469.748s	344.512s	172.377s	86.321s	43.300s	21.788s	11.030s	5.699s	3.049s

Data Transmission with the 3964/3964R Procedures

5

5

Contents of Chapter 5

Data Transmission with the 3964/3964R Procedures

5

5

You can use the second interface of the CPU 928B/CPU 948 as an interface for

- data transmission with the 3964/3964R procedures.

5.1 Introduction to 3964/3964R

Using the 3964/3964R procedure, data can be exchanged between two communications partners connected via a point-to-point link. Data transmission with the 3964/3964R procedure ensures a high degree of transmission reliability, since the transmission with this procedure covers the following layers of the ISO/OSI model (ISO IS 7498):

- Physical layer (layer 1):
  This layer handles the physical transmission of the data bytes (physical characteristics, transmission speed ...)

- Data link layer (layer 2):
  The 3964/3964R procedure adds start and end characters to the bit stream. It is also capable of detecting the loss or impaired transmission of frames and to initiate repetitions.

The protocol of the procedure only ensures the reliable transmission of the data, not, however, the processing or evaluation of the transmitted data.

Note

Data transmission with the 3964/3964R procedures uses the same procedures as the RK 512 computer link (see Chapter 4).

If a message is acknowledged positively by the partner (received and accepted), the message has been correctly received. This does not mean that the message is processed by the receiver (e.g. the data cannot be written to the receive mailbox by the system program, if the receive mailbox is too short).

If you want to make sure that the message is correctly processed, then you must either use the RK 512 computer link or take your own steps to ensure data protection.

This introduction contains basic information about data transmission with the 3964/3964R procedures, as follows:

- application
- jobs
- user program

5.1.1 Application

Data transmission with the 3964/3964R procedures implements data exchange between the CPU 928B/CPU 948 in an S5-135U or S5-155U and one of the following partners:

- a futher **CPU 928B** or **CPU 948** in the S5-135U and S5-155U programmable controllers
- a **CP 544** in the S5-115U/H, S5-135U and S5-155U/H programmable controllers
- a **non-SIMATIC** computer operating with the same procedure.

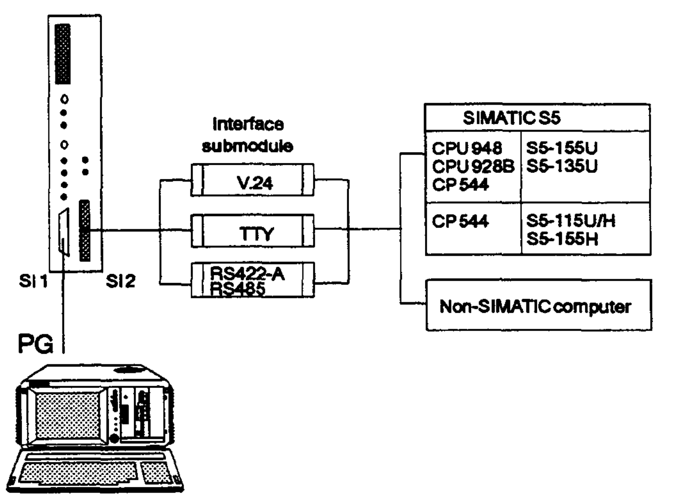


Fig. 5-1 Data exchange

Various interface submodules are available for data transmission using the 3964/3964R procedures, as follows:

- V.24 submodule
- TTY submodule
- RS422-A/485 submodule (only in the RS422-A mode)

Install the submodule you require for your application in the submodule receptacle of the second interface of the CPU 928B or CPU 948 (see Chapter 2, "Hardware Components"). You establish the link to the partner using the standard connecting cable. If you cannot use the standard connecting cables, and prepare your own cable, make sure that you keep to the pin assignments explained in Section 5.8 of the S5-135U/155U System Manual.

Set the transmission speed for the data exchange as suitable for the communications partner (see Section 5.2.2). The following transmission speeds are possible:

- 110 bps to 19 200 bps with the RS422-A/485 submodule
- 110 bps to 19 200 bps with the V.24 submodule
- 110 bps to 9 600 bps with the TTY submodule.

5.1.2 Jobs

With the 3964/3964R procedures, you can use the following active job:

- send data: **SEND job**
  the data to be sent are in the transmit mailbox (see Section 5.2.3).

With the 3964/3964R procedures, you can use the following passive job:

- receive data: **SEND job on the partner**
  the received data are in the receive mailbox (see Section 5.2.4).

A SEND job and a SEND job on the partner form a pair. A SEND job on the CPU 928B/CPU 948 leads to a SEND job on the partner and vice-versa.

SEND job

The CPU 928B and the CPU 948 are active, they send data to the partner. The partner is passive, it receives data.

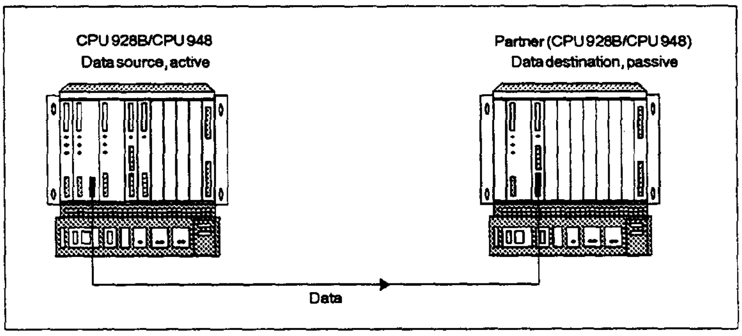


Fig. 5-2 Example of a SEND job with a second CPU 928B/CPU 948 as communication partner

SEND job on partner

The partner is active, it sends data to the CPU 928B/CPU 948. The CPU 928B/CPU 948 is passive, it receives data.

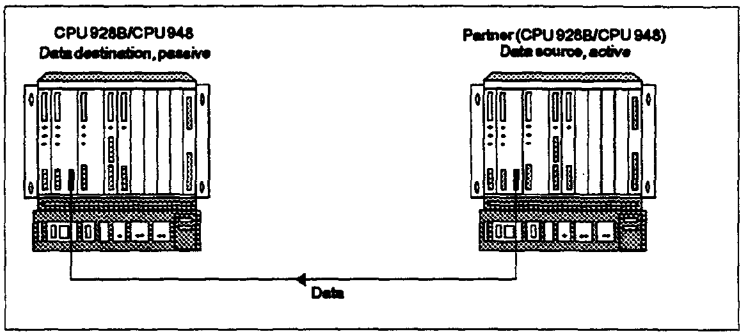


Fig. 5-3 Example of a SEND job on the partner with a second CPU 928B/ CPU 948 as communication partner

5

5.1.3 Incorporating the Procedures into the User Program

This section provides you with an overview of how to incorporate data transmission with the 3964/3964R procedures into your user program.

When creating your user program, you should follow the steps explained below:

Prior considerations

- Which devices will the CPU 928B or the CPU 948 exchange data with?
- Which interface submodule do you require?
- Which data will be exchanged?
- What will you call the jobs?

Assigning parameters to the interface

You must assign the parameters for the DX 2, a static parameter set, a transmit mailbox, a receive mailbox, the send coordination byte (SCB) and the receive coordination byte (RCB).

DX 2

You must enter the following information into DX 2:

- link type: data transmission with the 3964/3964R procedures
- location of the static parameter set
- location of the transmit mailbox
- length of the transmit mailbox
- location of the receive mailbox
- length of the receive mailbox
- location of the send coordination byte (SCB) and receive coordination byte (RCB)

Static parameter set

In the DB/DX containing the static parameter set you must enter the parameters for the physical layer and for the data link layer.

The parameters for the physical layer (layer 1) are as follows:

- transmission speed (bps)
- parity
- bits per character
- stop bits

The parameters of the data link layer (layer 2) that determine the characteristics of the 3964 or 3964R transmission procedure are as follows:

- mode: 3964/3964R procedure with defaults or selected values
- priority
- character delay time
- acknowledgement delay time
- connection establishment attempts
- number of repetitions

Transmit mailbox

The DB/DX with the transmit mailbox contains the data to be sent. The transmit mailbox can be a maximum of 2049 words long.

Receive mailbox

The DB/DX with the receive mailbox contains the received data. The receive mailbox can be a maximum of 2049 words long.

SCB

With the send coordination byte (SCB) you can perform the following:

- start a SEND job
- reset the data transmission with the 3964/3964R procedures

RCB

With the receive coordination byte (RCB) you can perform the following:

- enable and disable the receive mailbox

You can generate the parameter set in DX2 and the static parameter set either via the **DB editor** or via the menus in the **COM PP Parameter Assignment Software** (see Chapter 3).

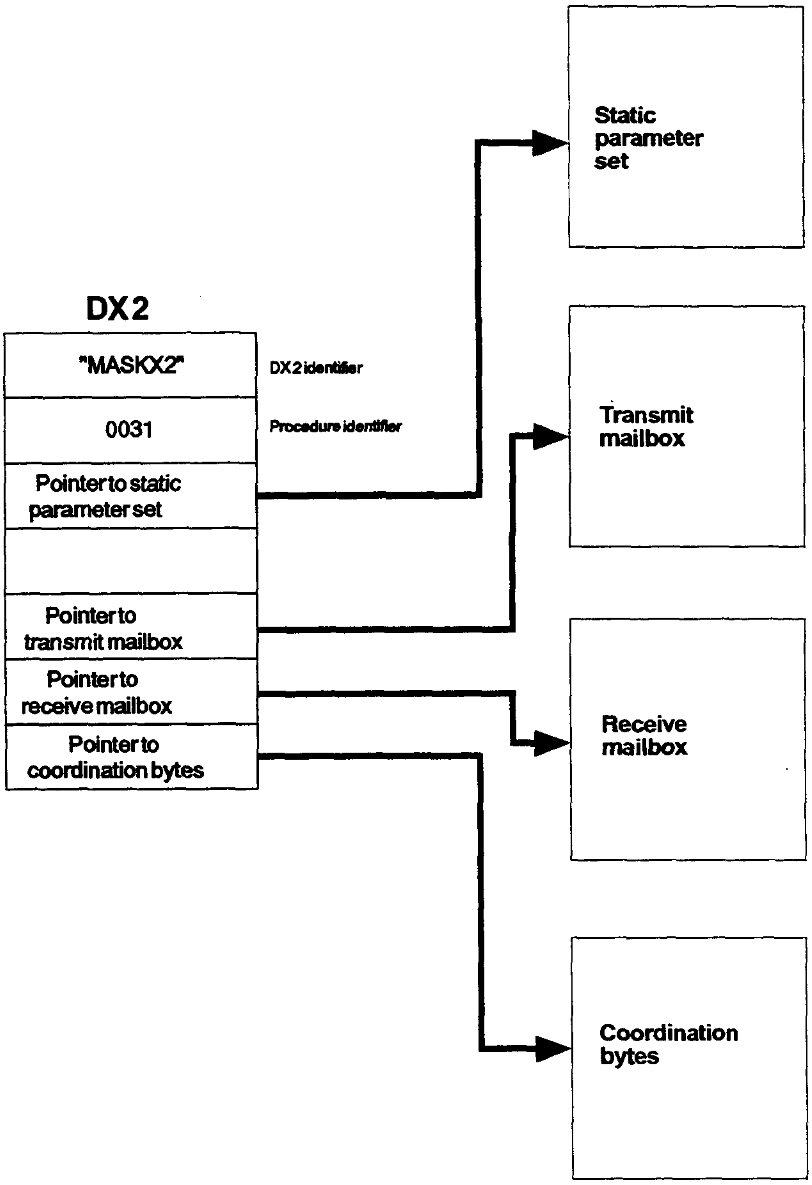


Fig. 5-4 Components of the user program

Operating your CPU 928B/ CPU 948 only as an active device

If you operate your CPU 928B or CPU 948 only as an **active** device that does not receive any data itself, you do not need a receive mailbox for data transmission with the procedures 3964/3964R.

Operating your CPU 928B/ CPU 948 only as a passive device

If you operate your CPU 928B or CPU 948 only as a **passive** device that does not trigger any jobs itself, you do not need a transmit mailbox for data transmission with the procedures 3964/3964R.

Note

If no DX 2 is available in the CPU, initialize the second interface as the PG interface.

Note

You must specify the parameters in DX 2 and in the static parameter set before you start up data transmission with the 3964/3964R procedures.

With the CPU 928B the parameters in DX 2 are only adopted by the system program following a cold restart.
With the CPU 948 the parameters are also adopted by the system program following power up or an overall reset (loading DX 2 from memory card).

The parameters in the static parameter set (DW 0 to DW 4) are only adopted by the system program following power up or a cold restart.

The parameters in the static parameter set from DW 5 onwards are only adopted by the system program following power up, cold restart and "reset data transmission with the 3964/3964R procedures".

You can only modify the content of the transmit mailbox and the SCB when no SEND job is active.

You can only modify the content of the receive mailbox and RCB when no SEND job is active on the partner.

The following chapters provide more detailed information about the values you can use when assigning parameters for data transmission with the 3964/3964R procedures.

5.2 Parameter Assignment

The following sections explain how you can assign parameters for data transmission with the 3964/3964R procedures and how you should structure the required data blocks.

You have two options:

- either assign parameters directly in STEP 5 by inputting them into data blocks
- or enter the parameters via the menus in the COM PP Parameter Assignment Software and then transfer the data blocks you have created to the CPU using the STEP 5 basic package. Working with COM PP is described in Chapter 3.

5.2.1 Assigning Parameters to DX 2

DX 2 contains the link type. It also contains the pointers to the required parameters and parameter sets. The following table shows the values you can use when assigning parameters to DX 2. The assignment in DX 2 begins at DW 0. All the numerical values are shown in hex.

DW	Parameter	Significance
0	4D41	MA
1	534B	SK
2	5832	X2
Link type		
3	0031	Data transmission with 3964/3964R
Pointer to static parameter set		
4	44xx or 58xx	DB no. xx or[4] DX no. xx
5	xxxx	from DW no. xxxx
-		
6	0000	reserved
7	0000	reserved
Pointer to transmit mailbox [1]		
8	44xx or 58xx	DB no. xx or[4] DX no. xx
9	xxxx	from DW no. xxxx
10	xxxx	Length of the transmit mailbox in words (range of values 0002H to 0801H)
Pointer to receive mailbox [2]		
11	44xx or 58xx	DB no. xx or[4] DX no. xx
12	xxxx	from DW no. xxxx
13	xxxx	Length of the receive mailbox in words (range of values 0002H to 0801H)
Pointer to SCB/RCB [3]		
14	4D00 or 5300 or 44xx or 58xx	Flag or[4] S flag or DB no. xx or DX no. xx
15	xxxx	FW no., SW no. or DW no. xxxx

1) These parameters are not necessary if the CPU 928B or the CPU 948 is purely passive. In this case, enter the value 0000 in data words DW 8 to DW 10

2) These parameters are not necessary if the CPU 928B or the CPU 948 is purely active, i.e. does not receive any data. In this case, enter the value 0000 in data words DW 11 to DW 13

3) The location of SCB and RCB must be fixed as a pair. For a detailed description, refer to Section 5.2.5.

4) Range of values for xx from 3 to FF (3 - 255)

Example

In this example, you can see how to assign the following basic parameters to a DX 2.

- The static parameter set is in data block 10 from data word 4 onwards.
- The transmit mailbox is in data block 6 from data word 2 onwards, length 10 words.
- The receive mailbox is in data block 8 from data word 1 onwards, length 6 words.
- The send coordination byte (SCB) and receive coordination byte (RCB) are in flag word 6.

With these parameters, DX 2 appears as below:

DX 2

0 :	KH = 4D41;	'MASKX2'
1 :	KH = 534B;	
2 :	KH = 5832;	
3 :	KH = 0031;	**LINK TYPE** 3964/3964R PROCEDURE
4 :	KH = 440A;	**STATIC PARAMETER SET** IN DB 10
5 :	KH = 0004;	FROM DATA WORD 4
6 :	KH = 0000;	reserved
7 :	KH = 0000;	
8 :	KH = 4406;	**TRANSMIT MAILBOX** IN DATA BLOCK 6
9 :	KH = 0002;	FROM DATA WORD 2
10:	KH = 000A;	LENGTH 10 WORDS
11:	KH = 4408;	**RECEIVE MAILBOX** IN DATA BLOCK 8
12:	KH = 0001;	FROM DATA WORD 1
13:	KH = 0006;	LENGTH 6 WORDS
14:	KH = 4D00;	**COORDINATION BYTES** IN FLAG AREA
15:	KH = 0006;	SCB FLAG BYTE 6, RCB FLAG BYTE 7
16:	KH = xxxx;	no longer belongs to the parameter assignment for DX 2, content irrelevant, does not need to exist.
:		
:		

5.2.2 Static Parameter Set

In the static parameter set, you must enter the parameters for the physical and data link layers.
The following table shows the values that you can enter in a data block or extended data block for the static parameter set. The parameter assignment begins at the DW specified in the DX 2.

All numerical values are shown in hexadecimal format.

DW (rel.)	Range of values	Significance
Transmission speed		
0	0001	110 bps
	0002	150 bps
	0003	300 bps
	0004	600 bps
	0005	1200 bps
	0006	2400 bps
	0007	4800 bps
	0008	9600 bps
	0009	19200 bps (only with V.24 submodule and RS422-A/485 submodule)
Parity		
1	0000	no
	0001	odd
	0002	even
Bits per character		
2	0006	6 bits per character
	0007	7 bits per character
	0008	8 bits per character
Stop bits		
3	0001	1 stop bit
	0002	1 + 1/2 stop bits
	0003	2 stop bits
-		
4	0000	reserved
Mode (see note)		
5	0001	3964 procedure with default values
	0002	3964R procedure with default values
	0003	3964 procedure with selectable values
	0004	3964R procedure with selectable values
Priority		
6	0000	low
	0001	high

DW (rel.)	Range of values	Significance
Character delay time (see note)		
7	0001 to FFFF	monitoring time 0.01 sec to 655.35 sec (unit: 10 ms)
Acknowledgement delay time (see note)		
8	0001 to FFFF	monitoring time 0.01 sec to 655.35 sec (unit: 10 ms)
Connection attempts (see note)		
9	0001 to 00FF	number
Repetitions		
10	0001 to 00FF	number

Note

Data words 7 to 10 are only evaluated if you enter 0003 or 0004 as the mode.

Caution

If you enter 0001 or 0002 as the mode, the default values are used. Remember, however, that data words 7 to 10 must nevertheless exist.

The standard values for mode 0001and mode 0002 are as follows:

	Mode 0001	Mode 0002
Character delay time	220 ms	220 ms
Acknowledgment delay time	550 ms	2000 ms
Connection attempts	6	6
Repetitions	6	6

Meaning of the parameters

Transmission speed

Specified in bps

Parity

Number of bits with the value "1" in a string of information bits with a uniform length (e.g. 1 byte). Depending on the value, the parity is "even" or "odd". To check the parity, the string of information bits is extended by a further bit, the parity bit ("0" or "1") which when added to the information bits produces a selectable parity state.

Bits per character

Number of data bits used to form a character.

Stop bits

Duration of the stop bits, relative to the time required to transmit an information bit. The stop bits follow each character transmitted in a start-stop transmission.

Mode

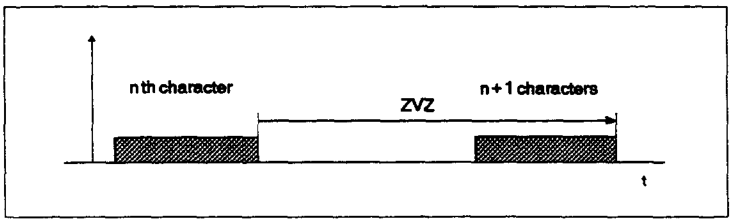
Version of the two possible transmission procedures, 3964 or 3964R, either with default values or with selected values.

Priority

A partner has high priority when its transmit request has priority over the transmit request of another partner. A partner has low priority when its transmit request must give way to the transmit request of the other partner. When using the computer link, the communications partners must have different priority, i.e. A has priority = 1 (high), B must have priority = 0 (low).

Character delay time (ZVZ)

The maximum permitted time interval between two received characters (see following table).



Acknowledgement delay time (QVZ)

The maximum permitted time before receiving the acknowledgement from the partner during connection establishment/termination (see following table).

Transmission speed	Min. permitted ZVZ	Min. permitted QVZ
110 bps	120 ms	120 ms
150 bps	100 ms	100 ms
300 bps	60 ms	60 ms
600 bps	40 ms	40 ms
1200 bps	30 ms	30 ms
2400 bps	20 ms	20 ms
4800 bps	20 ms	20 ms
9600 bps	20 ms	20 ms
19200 bps	20 ms	20 ms

Connection attempts

Maximum number of attempts made by the CPU 928B or CPU 948 to establish a connection.

Repetitions

Maximum number of message repetitions (including the first message) if errors occur.

Example

In this example, you can see how to assign values for a static parameter set in DB 10 from DW 4 onwards.

DB 10

0:	KH = xxxx;	does not belong to the static parameter set, content irrelevant
1:	KH = xxxx;	
2:	KH = xxxx;	
3:	KH = xxxx;	
4:	KH = 0008;	**TRANSMISSION SPEED** = 9600 bps
5:	KH = 0002;	**PARITY** = 2 = EVEN
6:	KH = 0007;	**BITS PER CHARACTER** = 7
7:	KH = 0001;	**STOP BITS** = 1
8:	KH = 0000;	reserved
9:	KH = 0003;	**MODE** = 3 = 3964 procedure (selectable values)
10:	KH = 0001;	**PRIORITY** = 1 = HIGH
11:	KH = 00FF;	**CHAR. DELAY TIME** = 255 x 0.01 sec (=2.55 sec)
12:	KH = 010A;	**ACK. DELAY TIME** = 266 x 0.01 sec (=2.66 sec)
13:	KH = 0003;	**ATTEMPTED CONNECTIONS** = 3
14:	KH = 0005;	**REPETITIONS** = 5
15:	KH = xxxx;	no longer belongs to the static parameter set, content irrelevant, does not need to exist
:		
:		

When transmitting data using the 3964/3964R procedures, the parameters for the transmission speed must be the same in the partner and in the CPU 928B/CPU 948.

The parameters for the data link layer in the partner must be set as follows:

- use of the same procedure as in the CPU 928B/CPU 948
- opposite priority from that of the CPU 928B or CPU 948

5.2.3 Transmit Mailbox

The transmit mailbox is in a data block (DB) or extended data block (DX).

The transmit mailbox contains the data to be transmitted and in the first word the specified length (in bytes) of the data to be transmitted. For each job, you can decide how many data are to be transmitted to the communications partner. The length does not include the word containing the length itself. The length specification is not transmitted.

The length of the transmit mailbox is a maximum of 2049 words (4098 bytes). Stipulate this in DX 2 (in words). Remember that the length of the data to be transmitted (in bytes) in the first word of the transmit mailbox must not be longer than the length (in words) of the transmit mailbox assigned in DX 2 minus the first word containing the length itself. Assign a transmit mailbox length suitable for your current task.

Example: transmit mailbox

In this example, you can see the structure of a transmit mailbox in data block 6 from data word 2 onwards with a length of 10 words. Seven bytes will be transmitted.

DB 6

0:	KH = xxxx;	does not belong to transmit mailbox, content irrelevant
1:	KH = xxxx;	
2:	KH = 0007;	length in bytes, start of transmit mailbox
3:	KH = 0102;	data to be transmitted: 0102
4:	KH = 0304;	data to be transmitted: 0304
5:	KH = 0506;	data to be transmitted: 0506
6:	KH = 07yy;	data to be transmitted: 07, yy is not transmitted
7:	KH = yyyy;	content of the transmit mailbox, is not transmitted here
8:	KH = yyyy;	
9:	KH = yyyy;	
10:	KH = yyyy;	
11:	KH = yyyy;	
12:	KH = xxxx;	does not belong to transmit mailbox, content irrelevant, does not need to exist
:		
:		

5.2.4 Receive Mailbox

The receive mailbox is in a data block (DB) or extended data block (DX).

The receive mailbox contains the data received and in the first word the specified length (in bytes) of the received data. This is entered by the system program, so that you can see how many bytes were received. The length does not include the word containing the length itself.

The length of the receive mailbox is a maximum of 2049 words (4098 bytes). Stipulate this in DX 2 (in words). You can only receive as many bytes as you specify in DX 2 as the length (in words) of the receive mailbox minus the first word in the receive mailbox containing the length specification.

Example: receive mailbox

In this example, you can see the structure of a receive mailbox in data block 8 from data word 1 onwards with a length of 6 words. Seven bytes were received.

DB 8

0:	KH = xxxx;	does not belong to receive mailbox, content irrelevant
1:	KH = 0007;	length in bytes, start of receive mailbox
2:	KH = 0102;	received data: 0102
3:	KH = 0304;	received data: 0304
4:	KH = 0506;	received data: 0506
5:	KH = 07yy;	received data: 07, yy was not received
6:	KH = yyyy;	content of receive mailbox was not received
7:	KH = xxxx;	does not belong to receive mailbox, content irrelevant, does not need to exist
:		
:		

5.2.5 The Send and Receive Coordination Bytes (SCB, RCB)

With the SCB you can start the following jobs:

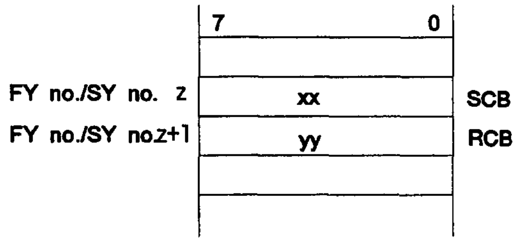
- SEND job
- reset the data transmission using 3964/3964R procedure

With the RCB you can perform the following:

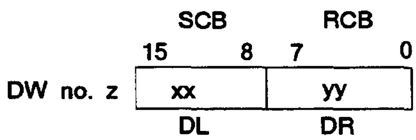
- enable and disable the receive mailbox

The coordination bytes can be flags or S flags or be located in a data block or extended data block. Flags and S flags already exist in the CPU 928B or in the CPU 948, you must set up data blocks and extended data blocks yourself.

Representation in the flag or S flag area

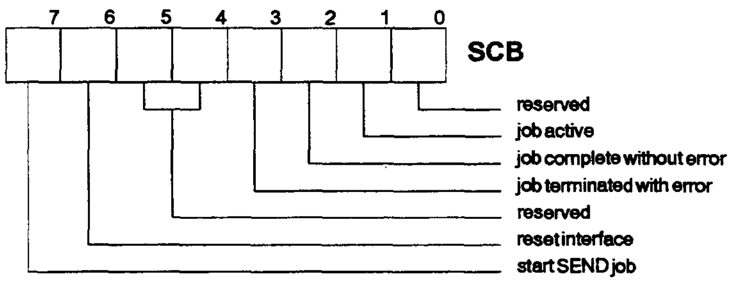


Representation in a data block or extended data block



Send coordination byte (SCB)

The send coordination byte is structured as follows:



The individual bits or bit groups, providing they are not reserved, are described below.

Bit 1 "Job active"

- set by the system program when a job starts
- reset by the system program when a job is completed
- you can read this bit, to know whether there is any point in starting a new job

Bit 2 "Job complete without error"

- set by the system program when a job is completed without error
- you can read this bit and then reset it, or it is reset by the system program when bit 1 "job active" is set for a new job
- you can read this bit, to know whether a job was completed without error

Bit 3 "Job terminated with error"

- set by the system program when a job is terminated with an error
- you can read this bit and then reset it, or it is reset by the system program when bit 1 "job active" is set for a new job.
- you can read this bit, to know whether a job was completed with an error

Bit 6 "Reset interface"

- you can set this bit to reset the interface
- reset by the system program as soon as the system program resets the interface

Bit 7 "Start SEND job"

- you can set this bit to start a SEND or FETCH job
- reset by the system program as soon as the system program completes the SEND or FETCH job

When you reset the interface, the parameters of the data link layer in the static parameter set (from DW 5 onwards) are read in again:

- mode
- priority
- character delay time
- acknowledgement delay time
- connection attempts
- repetitions

If the SCB bits 6 and 7 are set simultaneously or if SCB bit 6 is set followed by SCB bit 7 at an interval ≤ 10 ms (CPU 928B) or ≤ 20 ms (CPU 948), the system first carries out the reset followed by the SEND job. If SCB bit 7 is set first followed by SCB bit 6 at an interval ≤ 10 ms (CPU 928B) or ≤ 20 ms (CPU 948), the sequence of the two jobs cannot be guaranteed. By resetting, you abort the currently active partner SEND job.

Note

You can only modify the parameters of the physical layer including the physical transmission characteristics (static parameter sets up to and including DW 4) using a cold restart on the CPU 928B or on the CPU 948. Resetting the interface is not sufficient.

Receive coordination byte (RCB)

As long as RCB bit 7 is reset, no received data are entered in the receive mailbox.

RCB

7	6 ... 0
	reserved

Bit 7:
0 = receive mailbox occupied
1 = receive mailbox free

Caution

In your user program, you must make sure that the received data are processed as quickly as possible and that the receive mailbox is released again by RCB bit 7.

If further data are received while the receive mailbox is disabled by RCB bit 7, the data remain in the dual-port RAM (see Section 1.4 "Fundamentals of Communication"), until the receive mailbox is released again by RCB bit 7.

Further received data can be buffered in the receive buffer of the communications processor as long as the receive mailbox is disabled. Any further data received are then discarded.

Caution

During a cold restart or power up the SCB is set to 00H (no job active) and the RCB to 80H (receive mailbox free).

5.3 Getting Started

Preparations

You have already installed the interface submodule in the CPU 928B or in the CPU 948 (see Chapter 2), the CPU is plugged into the PLC rack and the power supply is off.

Procedure

We recommend the following procedure for starting the data transmission with the 3964/3964R procedure:

Step	Action
1	Connect the CPU 928B/CPU 948 to the partner (with a standard connecting cable).
2	Switch on the power supply.
3	Transfer your user program to the CPU. The user program must contain the following **data blocks** or **extended data blocks** to be able to set the SI 2 interface for data transmission with the 3964/3964R procedure. DX 2 DB/DX with the static parameter set DB/DX with the transmit mailbox [1)] DB/DX with the receive mailbox [2)] DB/DX with the SCB and RCB (only if the SCB/RCB is located in a DB/DX) [3)]
4	Perform a **cold restart**.
5	Prepare the partner for data transmission with the 3964/3964R procedures.

1) Only when the CPU 928B/CPU 948 triggers active jobs itself
2) Only when the CPU 928B/CPU 948 receives passive jobs
3) If the SCB and RCB are a flag or S flag word, it already exists in the CPU 928B/ CPU 948 and you do not need to transfer it

5

CPU 928B

During a cold restart or following power up the system checks that the following exist:

- DX 2
- the static parameter set
- SCB and RCB

and the correct parameter assignment in DX 2

DX 2 exists and has correct parameters

- The system program initializes the SI 2 interface for data transmission with the 3964/3964R procedure
- The red LED SI 2 goes off as soon as the data transmission with the 3964/3964R procedure is ready for operation
- The system program resets SCB bit 7 (i.e. no SEND job active)
- The system program sets RCB bit 7 (i.e. receive mailbox enabled)

↓

Data transmission with the 3964/3964R procedures is ready for operation

DX 2 does not exist or does not have correct parameters

- The CPU remains in the "STOP" mode
- The red LED SI 2 is lit in addition to the STOP LED
- The system program outputs an error message to RS 3 and RS 4

CPU 948

During a cold restart or following power up the system checks that the following exist:

- DX 2
- the static parameter set
- SCB and RCB

and the correct parameter assignment in DX 2

DX 2 exists and has correct parameters

- The system program initializes the SI 2 interface for data transmission with the 3964/3964R procedure
- The red LED SI 2 goes off as soon as the data transmission with the 3964/3964R procedure is ready for operation
- The system program resets SCB bit 7 (i.e. no SEND job active)
- The system program sets RCB bit 7 (i.e. receive mailbox enabled)

↓

Data transmission with the 3964/3964R procedures is ready for operation

DX 2 does not exist or does not have correct parameters

- The system program does not initialize the SI 2 interface for data transmission with the 3964/3964R procedure
- The red LED SI 2 does not go off

5.4 Operation

Preparations

The SI 2 interface has parameters assigned for data transmission with the 3964/3964R procedures. The SI 2 LED is off. Data transmission with the 3964/3964R procedures is ready for operation.

Method

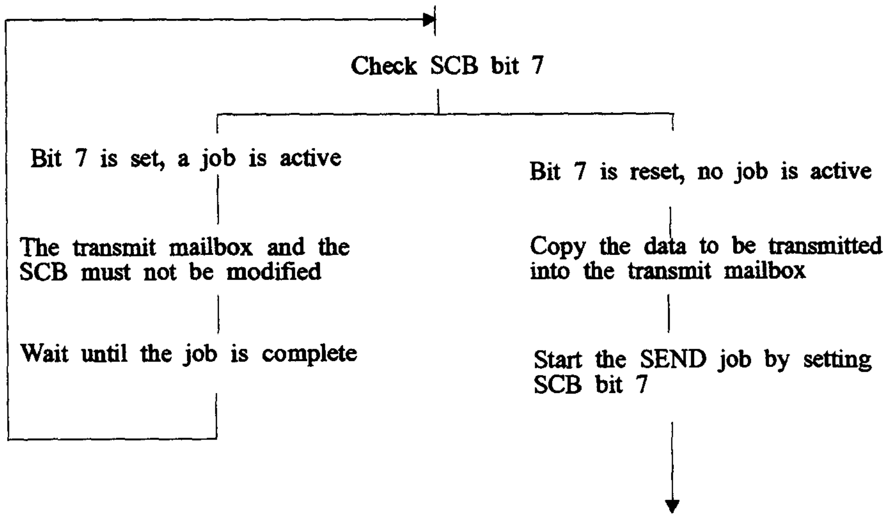
The system program now checks constantly whether the send and receive coordination bytes (SCB and RCB) exist. If they do not exist (e.g. because the DB/DX has been deleted), the system program calls OB 35. The error number is entered in ACCU 1 (see Chapter 11).

The system program now processes the jobs you have programmed in your user program, or the jobs triggered by the communications partner. The following sections explain how to program the jobs in your user program.

5.4.1 SEND Job

If you want to program a SEND job, remember the following sequence:

Start job:



The system program now processes your job.

The following diagram indicates the reactions that can occur if the SCB bit 7 is set and the system program is processing your job.

- SCB bit 7 is set
  - The transmit mailbox exists
    - The system program transfers the data specified by the length word (first word in the transmit mailbox). The length word is not transmitted.
      - The job was completed without errors
        - The system program
          - transferred the data from the transmit mailbox to the partner
          - reset the SCB bit 7
      - The job was completed with an error
        - The system program
          - called OB 35; ACCU 1 contains the error number
          - reset the SCB bit 7
  - The transmit mailbox does not exist
    - The system program
      - did not process the job
      - called OB 35 and entered an error number in ACCU 1
      - reset the SCB bit 7

Note

You can only start a new job when the previous job is completed.

5.4.2 Resetting the Interface

You have set the SCB bit 6 and therefore started a reset job. The following diagram shows the possible reactions.

SCB bit 6 is set

The CPU 928B/CPU 948 is currently processing a job from the partner

- the partner's job is aborted
- OB 35 has been called: ACCU 1 contains the error number for the aborted job of the partner
- the static parameters from DW 5 are read and evaluated
- when the reset job is complete, the SCB bit 6 is reset

The CPU 928B/CPU 948 is not processing a job from the partner

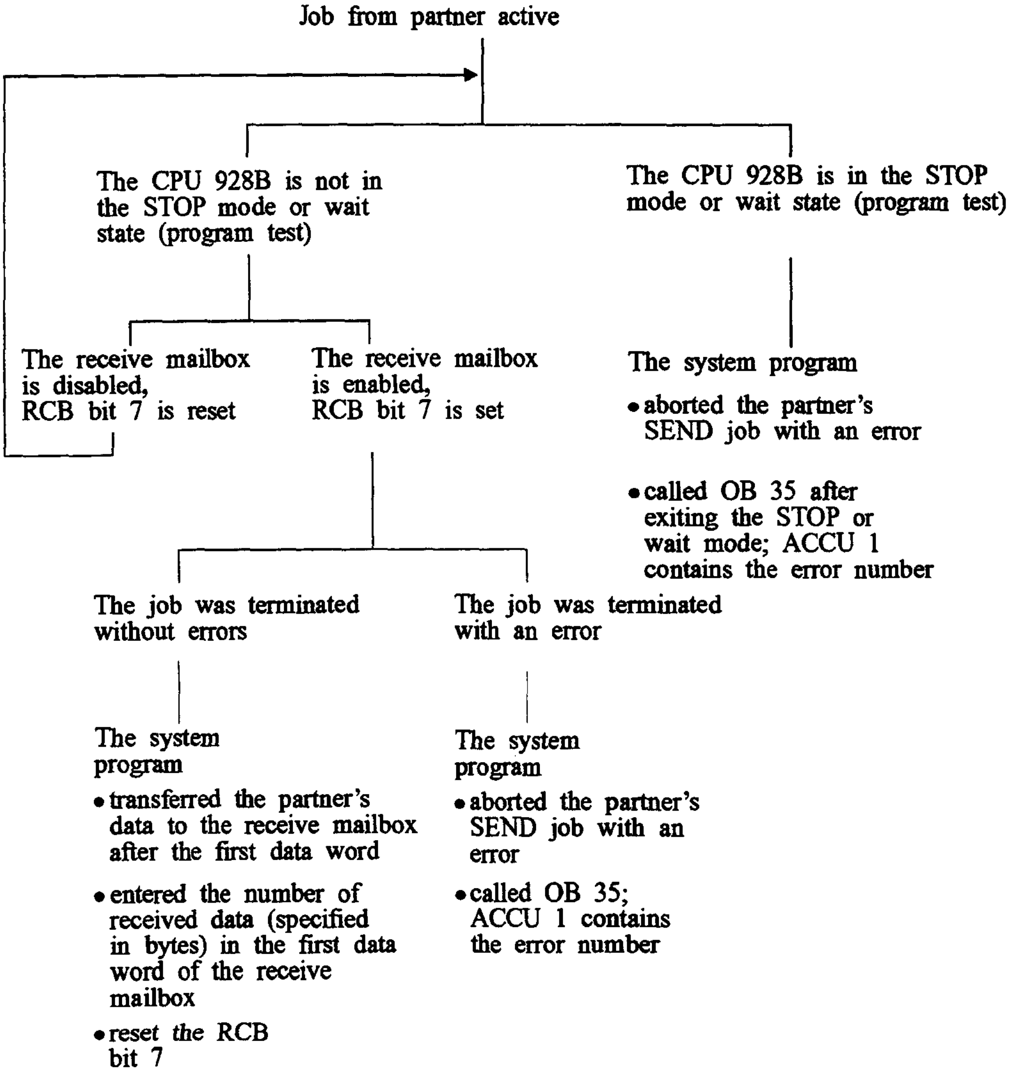
- the static parameters from DW 5 onwards are read and evaluated
- when the reset job is complete, the SCB bit 6 is reset

5.4.3 SEND Job on the Partner

SEND jobs on the partner are not processed by the user program but by the system program automatically in the background. The system program resets the RCB bit 7 (receive mailbox disabled), when the data have been completely transferred to the receive mailbox. You can now evaluate the receive data. The receive mailbox can only be overwritten again after it is released by RCB bit 7.

The following diagram shows situations that can occur following a job sent by the partner.

CPU 928B



CPU 948

Job from partner active

The receive mailbox is disabled, RCB bit 7 is reset

The receive mailbox is enabled, RCB bit 7 is set

The job was terminated without errors

The system program
- transferred the partner's data to the receive mailbox after the first data word
- entered the number of received data (specified in bytes) in the first data word of the receive mailbox
- reset the RCB bit 7

The job was terminated with an error

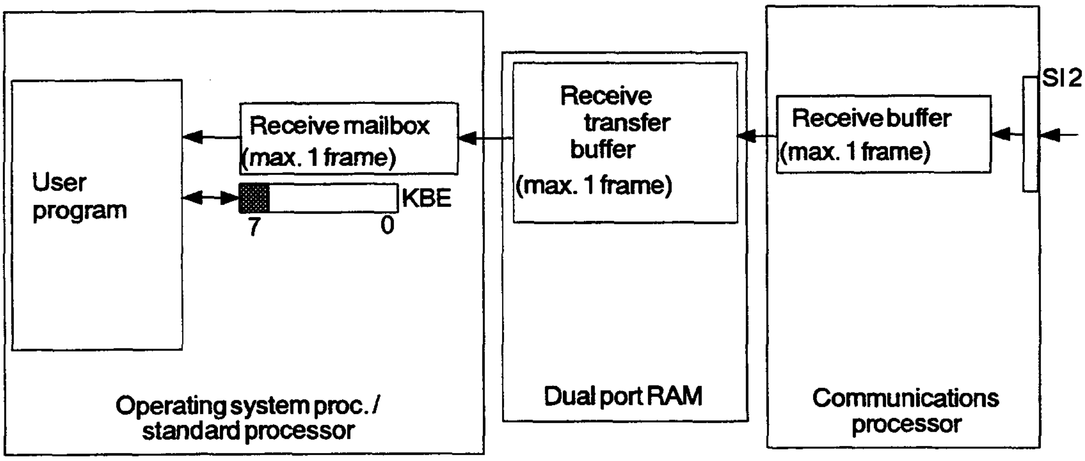
The system program
- aborted the partner's SEND job with an error
- called OB 35: ACCU 1 contains the error number

Caution

In your user program, you must make sure that the received data is processed as quickly as possible and that the receive mailbox is released again by RCB bit 7.

5.4.4 Points to Note when Receiving

Only one message can ever be entered in the receive mailbox (max. 4096 bytes long), although the receive transfer buffer of the dual-port RAM and the receive buffer can both hold one further message. Once you empty the mailbox and set bit 7 (receive mailbox free) in the receive coordination byte (RCB), the next message, if it exists, is transferred from the receive transfer buffer of the dual-port RAM into the mailbox and the next message shifted from the receive buffer to the receive transfer buffer of the dual-port RAM.



5.4.5 Response in Special Operating Modes

Stop mode

CPU 928B

In the "STOP" mode, user programs are not processed, i.e. you cannot start a new job. When the CPU changes from "RUN" to "STOP", the following applies to an already active job started by the user program:

- The job is stopped with an error, if there is still data to be transferred between the operating system processor and the dual-port RAM.
- The job is completed normally if there is no more data to be transferred between the operating system processor and the dual-port RAM.

If the partner triggers a job in the "STOP" mode, this is acknowledged negatively. An appropriate error message is entered in the error area of the interface (refer to Chapter 11 "Error Messages"). When the CPU changes from the "RUN" mode to the "STOP" mode, an already active job started by the partner is stopped with an error, i.e. negatively acknowledged. A corresponding error message is entered in the error area of the interface (see Chapter 11 "Error Messages")

CPU 948

In contrast to the CPU 928B, the CPU 948 also supports communication in "soft" stop and in the wait mode (program test).

Wait mode

CPU 928B

The CPU 928B goes into the wait mode when the PG online function "PROGRAM TEST" is active. In the wait mode user programs are not processed, i.e. you cannot start a new job. An already active job started by the user is continued and completed after the wait mode is exited.

If the partner triggers a job in the wait mode, this is acknowledged negatively. An appropriate error message is entered in the error area of the interface (refer to Chapter 11 "Error Messages"). When the CPU changes to the wait mode, an already active job started by the partner is negatively acknowledged. A corresponding error message is entered in the error area of the interface (see Chapter 11 "Error Messages")

CPU 948

In contrast to the CPU 928B, the CPU 948 also supports communicaiton in "soft" stop and in the wait mode (program test).

5.5 Procedures

The 3964 and 3964R procedures allow a simple data transmission in the quasi duplex mode with good transmission quality. These procedures are suitable not only for the CPU 928B or the CPU 948 but also for other units and computers from other manufacturers.

Control characters

Both procedures control the data flow between your CPU 928B or your CPU 948 and the partner. They are **asynchronous, bit serial transmission procedures.** The transmission and reception parameters (transmission speed, parity, ...) must be the same in the CPU 928B/CPU 948 and in the partner, the priority in the CPU 928B/CPU 948 and in the partner must be different.

Both procedures add **control characters** to the information characters when data is transmitted. With these control characters, the partner can check whether the data has been transmitted completely and without errors.

The CPU 928B/CPU 948 or partner transmits characters in bit serial form as follows:

SA	I0	I1	I2	I3	I4	I5	I6 [1]	I7 [2]	PA	SO

SA = start bit
I0...7 = information bits no. 0 to Nr. 7
PA = parity bit (if selected in the static parameter set)
SO = stop bit (length: 1, 1.5 or 2 bits)

[1] only exists if 7 or 8 bits per character is selected in the static parameter set
[2] only exists if 8 bits per character is selected in the static parameter set

The control characters for the procedures comply with the DIN standard 66003 for the 7-bit code. However, the character length (6, 7 or 8 bits) is used for transmission.

If the character DLE is transmitted as an information character, this is sent twice on the transmission line to distinguish it from the control character DLE used in connection establishment and termination (double DLE). The receiver "undoes" the doubling of the DLE.

No code is prescribed for other information characters (code transparency).

5.5.1 Transmitting

To establish the connection, the procedure sends the control character STX. If the partner replies with the DLE character before the acknowledgement delay time elapses (QVZ, for standard values, see Section 5.2.2) the procedure moves on to the transmission. If the partner replies with NAK, any other character (except for DLE) or does not reply within the acknowledgement delay time, the procedure repeats the connection establishment. After the number of unsuccessful connection attempts as specified in the static parameter set, the procedure stops attempting to establish the connection and sends the NAK character to the partner. The system program calls OB 35 and signals an error in connection establishment (the error number is located in ACCU 1).

If the connection is established successfully, the information data contained in the current output buffer are transmitted to the partner using the selected transmission parameters. The partner monitors the intervals between the incoming characters. The interval between two characters must not exceed the character delay time (ZVZ, for standard values see Section 5.2.2).

Double DLE

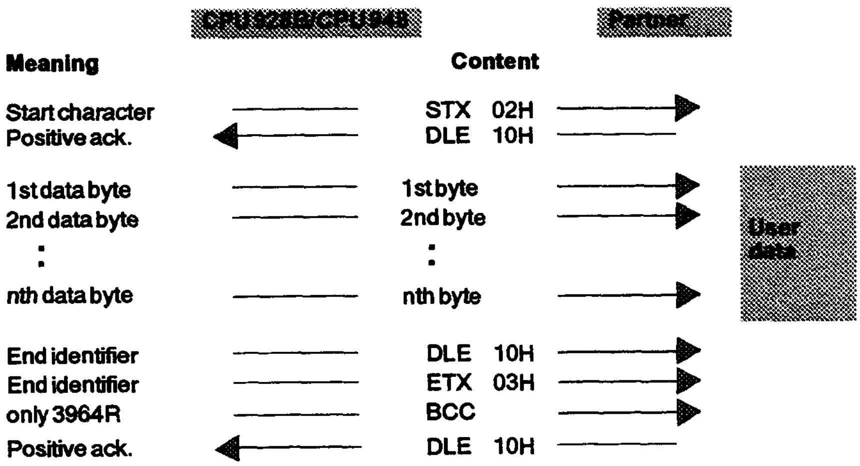
Each DLE character in the buffer is transmitted as two DLE characters (double DLE), i.e. each character \10H\ is sent twice. The receiver "undoes" the DLE doubling.

Block check character

After transmitting the content of the buffer, the procedure appends the characters DLE, ETX and **with 3964R only** also the block check character BCC as the end identifier and waits for an acknowledgement character. The block check character BCC is a byte. It is formed from the even longitudinal parity of all characters (EXOR logic operation), beginning with the first character after the establishment of the connection and finishing after the characters DLE ETX of the connection termination. The checksum is formed after the DLE doubling. If the partner sends the character DLE within the acknowledgement delay time (QVZ), the frame has been correctly received. If the partner replies with NAK, any other character (apart from DLE), an impaired character or the acknowledgement delay time elapses without a reaction, the procedure repeats the transmission with the connection establishment character STX. Once the connection has been attempted for the number of repetitions specified in the static parameter set, the procedure terminates the connection attempts and sends NAK to the partner. The system program calls OB 35 and signals the error (error number in ACCU 1).

If the partner sends the NAK character during a currently active transmission, the procedure terminates the frame and repeats it in the way described above. If any other character is transmitted, the procedure first waits until the character delay time has elapsed (ZVZ) and then sends NAK to bring the partner to the idle state. Following this, the procedure once again repeats the connection establishment with STX.

Example of an error-free data transmission:



5.5.2 Receiving

In the idle state, when there is no send job to be processed, the procedure waits for the connection to be established by the partner.

If the procedure receives any character in the idle state (apart from STX or NAK), it waits until the character delay time (ZVZ) has elapsed and then sends a NAK character. The system program calls OB 35 and signals the error (error number in ACCU 1).

Receive buffer

If the procedure receives the STX character and it has an empty receive buffer available, it replies with DLE. Incoming characters are now written to the receive buffer. If two consecutive DLE characters are received, only one DLE character is entered in the receive buffer. If the receive buffer is full, before the partner has started the connection termination procedure, the connection is terminated with NAK.

Following each received character, the procedure starts the character delay time (ZVZ) and waits for the next character. If the character delay time elapses without receiving another character, the NAK character is sent to the partner. The system program calls OB 35 and signals the error (error number in ACCU 1).

Reception free of errors/ errors detected

When the **3964** procedure recognizes the character string DLE ETX, it terminates reception and sends DLE to the partner if the frame was received free of errors or NAK if errors were detected. The procedure returns to the idle state.

If the **3964R** procedure detects the character string DLE ETX BCC, it terminates reception. It compares the received block check character with the internally generated longitudinal parity. If the block check character is correct and no other errors have occurred, the 3964R procedure sends DLE to the partner and returns to the idle state. If the BCC indicates an error or if there is any other error in reception, NAK is sent to the partner. Following this, the procedure expects a repetition.

If there is no empty receive buffer when the connection is established with STX, the procedure starts a wait time of 400 ms. If there is still no empty buffer after this time, the system program calls OB 35 and signals the error (error number in ACCU 1). The procedure sends the NAK character and returns to the idle state. Otherwise, the procedure sends the DLE character and receives the data as described above.

If transmission errors occur during reception (lost character, frame error, parity error etc.), the procedure continues to receive until the connection is terminated and then sends NAK to the partner. It then expects a repetition. If the frame cannot be transmitted correctly following the number of repetitions specified in the static parameter set or if the partner does not repeat the transmission within a waiting time of 4 seconds, the procedure terminates reception. The system program calls OB 35 and signals the error (error number in ACCU 1).

Example of error-free reception of data:

Meaning	CPU 928B/CPU 948	Content	Partner	
Start character	←	STX 02H		
Positive ack.		DLE 10H	→	
1st data byte	←	1st byte		User data
2nd data byte	←	2nd byte		User data
:		:		User data
nth data byte	←	nth byte		User data
End identifier	←	DLE 10H		
End identifier	←	ETX 03H		
only 3964R	←	BCC		
Positive ack.		DLE 10H	→	

Note

The 3964/3964R procedures send the NAK character to the partner following initialization or a reset to bring the partner to the idle state.

5.5.3 Initialization Conflict

If a device does not reply to the transmit request (STX character) of a partner within the acknowledgement delay time QVZ with the acknowledgement DLE or NAK, but with the character STX, there is an initialization conflict. Both devices want to execute their own transmit job. The device with the lower priority retracts its transmit job and replies with the DLE character. The device with the higher priority sends its data as described above. Following the connection termination, the device with the lower priority can execute its transmit job.

Priority

To deal with initialization conflicts, you can set the priority of the CPU 928B or of the CPU 948. The partner must then have the opposite priority.

The priority parameter is selected in the static parameter set (Section 5.2.2).

Example of handling an initialization conflict.

	CPU 928B/CPU 948 Low priority		Partner High priority	
Meaning		**Content**		
Start character		STX 02H	→	
Start character	←	STX 02H		
Positive ack.		DLE 10H	→	
1st data byte	←	1st byte		User data
2nd data byte	←	2nd byte		User data
:		:		User data
nth data byte		nth byte		User data
End identifier	←	DLE 10H		
End identifier	←	ETX 03H		
only 3964R	←	BCC		
Positive ack.		DLE 10H	→	
Start character		STX 02H	→	
Positive ack.	←	DLE 10H		
	:		:	

5.5.4 Procedure Errors

The procedure detects errors resulting from the incorrect response of the partner and errors caused by transmission impairment on the transmission line.

In both cases, the procedure first attempts to repeat the transmission or reception of the data. If the data cannot be transmitted or received free of errors and the maximum number of repetitions is reached (specified in the static parameter set) (or a different error occurs) the procedure terminates the transmission or reception. It then signals the error number of the first detected error and changes to the idle state. The system program calls OB 35, the error number is located in ACCU 1.

Interference

If the system program calls OB 35 often and enters error numbers in ACCU 1 for transmission and reception repetitions, you can assume that the data traffic is sporadically impaired. This can, however, be compensated by the number of repetitions. In this situation it is advisable to check the transmission line for interference, since the user data rate and reliability of the transmission is reduced by large numbers of repetitions. The cause of the problem can also result from the incorrect response of the partner.

Note

If BREAK is detected on the receive line, the system program indicates the BREAK state in OB 35 (ACCU 1). No repetition is started. There is no further activity until the BREAK state is cleared. OB 35 is only called again when another error occurs.

The same number is signalled for all detected transmission errors (lost character, frame error, parity error) regardless of whether the error was detected when transmitting or receiving a frame. The error is, however, only signalled when the previous repetitions were unsuccessful. If the impaired character is received while the procedure is in the idle state, the system program calls OB 35 and signals the error (in ACCU 1) to inform you of serious problems on the transmission line.

5.6 Processing Times

This section lists the typical processing times for a SEND job.

5.6.1 Processing Times for the CPU 928B

The processing times are subject to the following conditions:

- two CPU 928Bs are connected
- both CPUs have parameters assigned for the 3964 procedure in mode 1
  (even parity, 8 bits per character, 1 stop bit)
- the active CPU has the higher priority

The processing time is the time
- from starting the job on the active CPU 928B (the user program sets SCB bit 7)
- until the job is completed on the active CPU 928B (the system program resets SCB bit 7)

SEND job

The times were measured
- with different transmission speeds and
- with various amounts of user data

No. of bytes	Transmission speed								
	110	150	300	600	1200	2400	4800	9600	19200
1	0.644s	0.474s	0.239s	0.129s	0.069s	0.039s	0.020s	0.019s	0.009s
2	0.744s	0.547s	0.279s	0.139s	0.079s	0.039s	0.029s	0.019s	0.010s
4	0.944s	0.694s	0.349s	0.179s	0.089s	0.049s	0.029s	0.019s	0.019s
8	1.344s	0.989s	0.499s	0.249s	0.129s	0.069s	0.039s	0.029s	0.019s
16	2.146s	1.579s	0.789s	0.399s	0.209s	0.109s	0.059s	0.030s	0.019s
32	3.747s	2.749s	1.379s	0.689s	0.349s	0.179s	0.099s	0.049s	0.029s
64	6.950s	5.100s	2.550s	1.279s	0.649s	0.329s	0.169s	0.089s	0.049s
128	13.343s	9.788s	4.900s	2.459s	1.229s	0.619s	0.319s	0.159s	0.089s
256	26.148s	19.177s	9.594s	4.799s	2.409s	1.209s	0.609s	0.309s	0.159s
384	38.946s	28.560s	14.288s	7.149s	3.579s	1.799s	0.899s	0.459s	0.230s
512	51.748s	37.948s	18.979s	9.499s	4.750s	2.379s	1.199s	0.600s	0.309s
1024	102.960s	75.504s	37.760s	18.890s	9.459s	4.739s	2.379s	1.199s	0.609s
2048	205.376s	150.600s	75.316s	37.680s	18.860s	9.449s	4.739s	2.389s	1.209s
3072	307.792s	225.728s	112.886s	56.480s	28.267s	14.160s	7.108s	3.579s	1.819s
4096	410.240s	300.816s	150.440s	75.252s	37.666s	18.868s	9.468s	4.769s	2.419s

5.6.2 Processing Times for the CPU 948

The processing times are subject to the following conditions:

- two CPU 948s are connected
- both CPUs have parameters assigned for the 3964 procedure in mode 1
  (even parity, 8 bits per character, 1 stop bit)
- the active CPU has the higher priority

The processing time is the time

- from starting the job on the active CPU 948 (the user program sets SCB bit 7)
- until the job is completed on the active CPU 948 (the system program resets SCB bit 7)

SEND job

The times were measured
- with different transmission speeds and
- with various amounts of user data

No. of bytes	Transmission speed								
	110	**150**	**300**	**600**	**1200**	**2400**	**4800**	**9600**	**19200**
1	0.659s	0.489s	0.259s	0.139s	0.079s	0.049s	0.039s	0.029s	0.029s
2	0.757s	0.562s	0.289s	0.159s	0.089s	0.059s	0.039s	0.029s	0.029s
4	0.965s	0.719s	0.366s	0.199s	0.109s	0.069s	0.049s	0.039s	0.029s
8	1.361s	1.010s	0.519s	0.269s	0.149s	0.089s	0.059s	0.039s	0.029s
16	2.159s	1.590s	0.809s	0.419s	0.219s	0.119s	0.069s	0.049s	0.039s
32	3.758s	2.759s	1.393s	0.709s	0.369s	0.199s	0.109s	0.069s	0.049s
64	6.960s	5.113s	2.569s	1.297s	0.659s	0.339s	0.182s	0.109s	0.069s
128	13.364s	9.804s	4.914s	2.469s	1.249s	0.636s	0.329s	0.179s	0.099s
256	26.164s	19.192s	9.609s	4.818s	2.419s	1.219s	0.629s	0.329s	0.179s
384	38.965s	28.581s	14.303s	7.166s	3.599s	1.809s	0.919s	0.469s	0.249s
512	51.770s	37.967s	18.997s	9.509s	4.769s	2.399s	1.209s	0.619s	0.319s
1024	102.982s	75.522s	37.778s	18.909s	9.469s	4.749s	2.389s	1.209s	0.620s
2048	205.408s	150.631s	75.339s	37.692s	18.869s	9.459s	4.749s	2.399s	1.229s
3072	307.836s	225.7242s	112.900s	56.479s	28.269s	14.166s	7.119s	3.589s	1.829s
4096	410.262s	300.851s	150.463s	75.269s	37.669s	18.878s	9.479s	4.779s	2.429s

Data Transmission with the "Open Driver"

6

Contents of Chapter 6

6 Data Transmission with the "Open Driver"

You can use the second interface of the CPU 928B/CPU 948 as an interface for

- data transmission with the "open driver".

6.1 Introduction to the "Open Driver"

Using the open driver, you can exchange data between two communications partners on a point-to-point link using the duplex mode. Transmission with the "open driver" covers the following layer of the ISO/OSI model (ISO IS 7498):

- Physical layer (layer 1):
  this layer handles the physical transmission of data bytes (physical characteristics, transmission speed, ...)

Note

When using the "open driver" you must implement data protection measures yourself. The "open driver" does not guarantee that a message

- is received free of errors or
- is processed by the partner.

If you wish to make sure that a message is received free of errors, you must

- use data transmission with the 3964/3964R procedures.

If you want to make sure that a message is received free of errors and processed by the partner, you must

- either use the RK 512 computer link or
- provide data protection measures yourself for the user data.

This introduction contains basic information about data transmission with the "open driver", as follows:

- application
- transmission
- jobs
- user program

6.1.1 Application

Data transmission with the "open driver" implements the direct exchange of data between the CPU 928B/CPU 948 in an S5-135U or S5-155U and one of the following partners:

- another **CPU 928B/CPU 948** in the S5-135U and S5-155U programmable controllers
- a **CP 544** in the S5-115U/H, S5-135U and S5-155U/H programmable controllers
- an **input/output unit** operating with the same transmission procedure, for example a printer
- a **non-SIMATIC unit** operating with the same transmission procedure

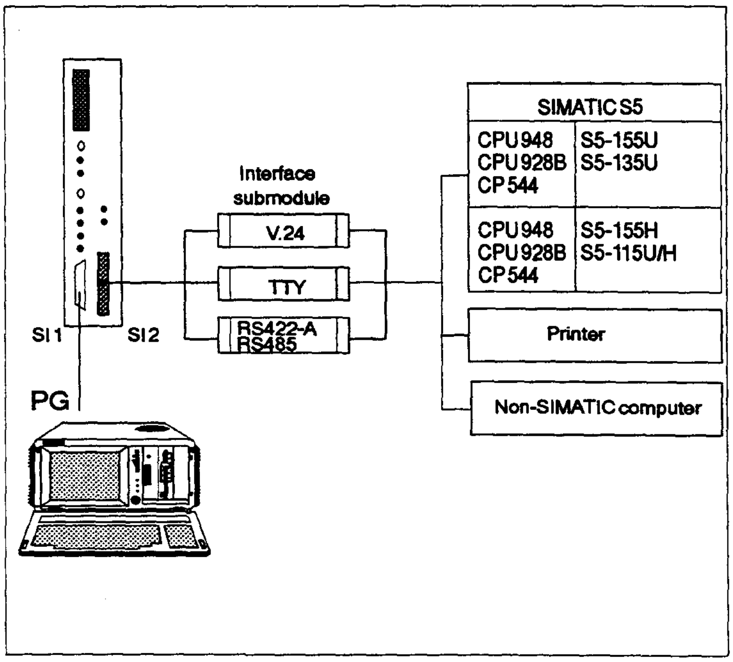


Fig. 6-1 Data exchange

Various interface submodules are available for the data transmission with the "open driver", as follows:

- V.24 submodule
- TTY submodule
- RS422-A/485 submodule (only in the RS422-A mode)

Install the submodule you require for your application in the submodule receptacle of the second interface of the CPU 928B/ CPU 948 (see Chapter 2, "Hardware Components"). You establish the link to the partner using a standard connecting cable. If you cannot use the standard connecting cables and prepare your own cable, make sure that you keep to the pin assignments explained in Section 5.8 of the S5-135U/155U System Manual.

If you use the V.24 submodule and use the RTS/CTS for flow control (hardware flow control), you must wire the interface signals in the connector completely.

For further information about the interface modules and standard connecting cables, refer to Section 5.8 of the S5-135U/155U System Manual.

Set the transmission speed for the data exchange as suitable for your application. The following transmission speeds are possible:

- 110 bps to 19 200 bps with the RS422-A/485 submodule
- 110 bps to 19 200 bps with the V.24 submodule.
- 110 bps to 9 600 bps with the TTY submodule.

Caution

Incorrectly received messages are discarded.

Note

Data transmission with the "open driver" does not correspond to DIN 66021 or the recommendations of CCITT V.24 in all aspects.

Remember the following **exceptions:**

The control signals M1/S1.1 (107/108.1 of the V.24 standard, DTR/DSR of RS232) are not generated or evaluated.

The control signals M2/S2 (106/105 for V.24, RTS/CTS for RS232) are only generated or evaluated as flow control if you select appropriate parameters (flow control = 2). These control signals do not correspond to DIN 66021 or the recommendations of CCITT V.24. They are used here for hardware flow control similar to XON/XOFF.

6.1.2 Transmission Mode

For data transmission with the "open driver" you can select between four different transmission modes. The mode is selected in the static parameter set.

Mode 1 (fixed user data length)

Sending

When sending n bytes of user data, you must specify the number n of user data to be transmitted in the first word of the transmit mailbox. This specified length does not include the word containing the length specification itself.

Receiving

When receiving n bytes of user data, the system program enters the number n of received useful bytes in the first word of the receive mailbox. This length does not include the word containing the length specification.

The end of a message is detected when the character delay time has elapsed.

Mode 2 (variable user data length)

Sending

When sending data, the user data including the end identifier (as defined in relative DW 11 and relative DW 12 of the static parameter set) is transferred from the transmit mailbox. In the first word of the transmit mailbox, you must specify the length (in bytes) of the user data relevant for the transmission. The end identifier must be part of this user data. The specified length does not include the word containing the length specification itself, which is not transmitted.

Receiving

When receiving data, the user data including the end identifier (as defined in relative DW 11 and relative DW 12 of the static parameter set) are entered in the receive mailbox after the first data word. The system program enters the number n (in bytes) of received user data bytes (including the end identifier) in the first word of the receive mailbox. This specified length does not include the word containing the length specification itself.

If the character delay time elapses during the reception of the user data, the reception is terminated. The message is discarded.

Mode 3 (fixed user data length, unsymmetrical)

Sending

When sending n bytes of user data, you must specify the number n of user data bytes to be transmitted in the first word of the transmit mailbox. This specified length does not include the word containing the length specification itself. The length specification is not transmitted.

Receiving

Even if you are only transmitting, the relative DW 13 must still be assigned a value in the static parameter set (0000H < DW 13 ≤ 1000H).

When receiving data, the user data is entered in the receive mailbox after the first data word up to the selected fixed message length (i.e. in the static parameter set, the relative DW 13 must be assigned a value). The system program enters the number n of received user data bytes in the first word of the receive mailbox (must correspond to the "fixed message length for reception"). This specified length does not include the word containing the length specification itself.

If the character delay time elapses during reception of the user data, the reception is terminated, the message is discarded.

Mode 4 (printer output)

Sending

When you output on a printer, the transmit mailbox up to and including the end identifier (as programmed in relative DW 11 and DW 12) is transmitted. In the first word of the transmit mailbox, you must specify the length (in bytes) of the user data relevant for the transmission. The end identifier is included in the user data. This specified length does not include the word containing the length specification itself. The length specification is not transmitted. Data words 14 to 23 ff. of the static parameter set are only relevant in this mode.
During the output, the data is prepared for printing (parameters from address 14H in the specific static parameter set). The print job is terminated after the end identifier has been sent, the line counter and similar values required for printing remain valid.

Receiving

Reception of characters does not take place in this mode. The only exception to this are the control characters for flow control if this has been selected (flow control 1). Any characters that may be received are discarded.

6.1.3 Jobs

With the "open driver" you can select the following in the modes 1 to 3 as active partner

- send data: **SEND job**
  The data to be sent are located in the transmit mailbox.

With the "open driver" you can select the following in mode 4 (printer output) as active partner

- transfer data to an output device (printer)
  For the CPU 928B or the CPU 948, this corresponds to a **SEND job**. The data to be sent are located in the transmit mailbox.

With the "open driver" you can select the following in modes 1 to 3 as passive partner

- receive data: **SEND job on the partner**
  The received data are located in the receive mailbox.

A SEND job and a SEND job on the partner form a pair. A SEND job on the CPU 928B or on the CPU 948 leads to a SEND job on the passive partner and vice versa.

SEND job (modes 1 to 3)

The CPU 928B/CPU 948 is active - it sends data to the partner. The partner is passive - it receives the data.

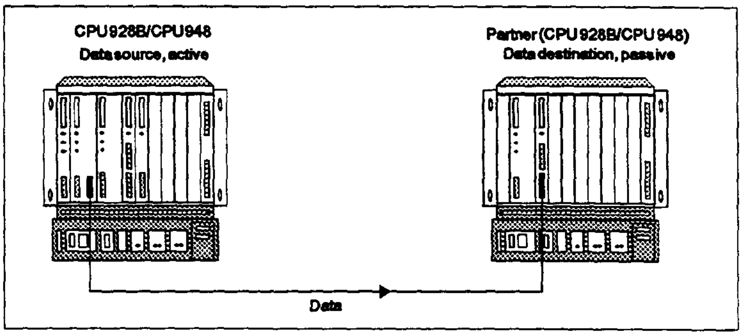


Fig. 6-2 Example of a SEND job with a second CPU 928B or CPU 948 as partner

SEND job (mode 4)

The CPU 928B/CPU 948 is active - it sends data to an output device. The output device is passive - it receives the data.

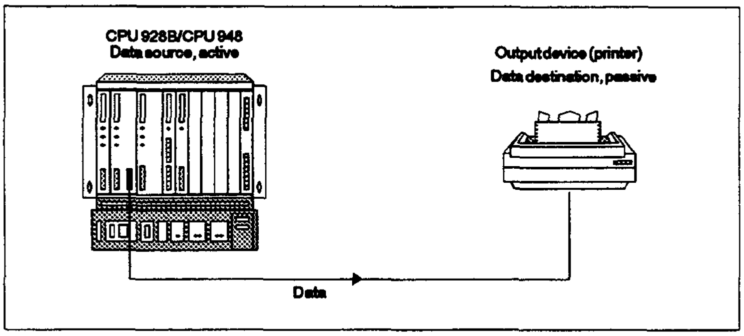


Fig. 6-3 Example of a SEND job in mode 4 with a printer as output device

SEND job on partner (modes 1 to 3)

The partner is active - it sends data to the CPU 928B or to the CPU 948. The CPU 928B/CPU 948 is passive - it receives the data.

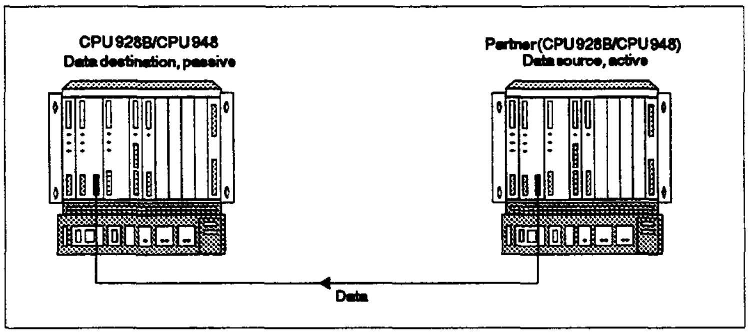


Fig. 6-4 Example of a partner SEND job with a second CPU 928B/ CPU 948 as partner

Note

The duplex mode (simultaneous transmission and reception) is supported. Data is transmitted in both directions (in the duplex mode) independent of the data terminal equipment and user program. A mode with an automatic echo of received characters as is known in half-duplex is not implemented.

6.1.4 Incorporation into the User Program

This section provides you with an overview of how to incorporate data transmission with the "open driver" into your user program.

When creating your user program, you should follow the steps explained below:

Prior considerations

- Which devices will the CPU 928B or the CPU 948 exchange data with?
- Which interface submodule do you require?
- Which data will be exchanged?
- What will you call the jobs?

Assigning parameters to the interface

You must assign parameters for **DX 2**, a static parameter set, a transmit mailbox, a receive mailbox, a send coordination byte (SCB) and a receive coordination byte (RCB).

6

DX 2

You must enter the following information in the extended data block DX 2:

- link type: data transmission with the "open driver"
- location of the static parameter set
- location of the transmit mailbox
- length of the transmit mailbox
- location of the receive mailbox
- length of the receive mailbox
- location of the send and receive coordination bytes (SCB and RCB)

Static parameter set

In the DB/DX containing the **static parameter set** you must enter the parameters for the physical layer.

The parameters of the physical layer (layer 1) are as follows:

- transmission speed (bps)
- parity
- bits per character
- stop bits

Further parameters in the static parameter set are as follows:

- flow control
- transmission mode
- character delay time
- end identifier
- fixed frame length for reception
- parameters for printer output

Which of these parameters are required in which transmission mode is explained in Sections 6.2 to 6.5.

Transmit mailbox

The DB/DX with the transmit mailbox contains the data to be transmitted. The transmit mailbox can be a maximum of 2049 words long.

Receive mailbox

The DB/DX with the receive mailbox contains the received data. The receive mailbox can be a maximum of 2049 words long.

SCB

With the send coordination byte (SCB) you can start the following jobs:

- a SEND job
- reset data transmission with the "open driver"

RCB

With the receive coordination byte (RCB) you can perform the following:

- enable and disable the receive mailbox

You can generate the parameter set in DX 2 and the static parameter set either via the **DB editor** or via the menus in the **COM PP Parameter Assignment Software** (see Chapter 3).

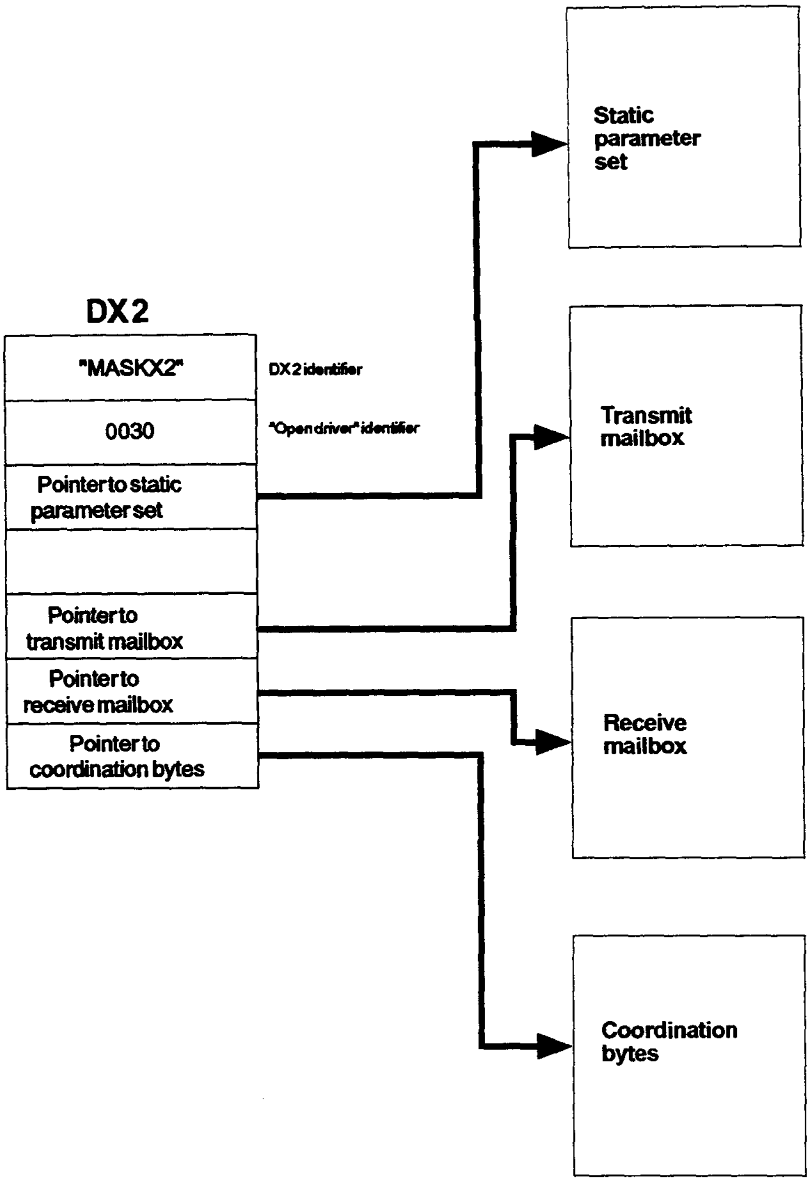


Fig. 6-5 Components of the user program

Operating your CPU 928B/ CPU 948 solely as a passive device

If you only wish to operate your CPU 928B or your CPU 948 as a **passive** device, that does not trigger jobs itself, you do not need a transmit mailbox for data transmission with the "open driver".

Operating your CPU 928B/ CPU 948 solely as an active device

If you only wish to operate your CPU 928B or your CPU 948 as an **active** device, that does not receive data itself, you do not need a receive mailbox for data transmission with the "open driver".

Operating your CPU 928B/ CPU 948 in mode 4 (printer output)

If you select mode 4 for your CPU 928B or for your CPU 948, you do not need a receive mailbox for data transmission with the "open driver".

Note

If no DX 2 is available in the CPU, initialize the second interface as the PG interface.

Note

You must specify the parameters in DX 2 and in the static parameter set before you start data transmission with the "open driver".

With the CPU 928B, the parameters in DX 2 are only adopted by the system program following a cold restart.
With the CPU 948, the parameters are also adopted by the system program following a power up or an overall reset (loading DX 2 from memory card).

The parameters in the static parameter set (DW 0 to DW 4) are only adopted by the system program following power up or a cold restart.

The parameters in the static parameter set from DW 5 onwards are only adopted by the system program following power up, cold restart and the job "reset data transmission with the open driver".

You can only modify the content of the transmit mailbox and the SCB when no job is currently active.

The following sections provide you with more detailed information about the values you can use when assigning parameters for data transmission with the "open driver".

6.2 Mode 1 - Fixed User Data Length

6.2.1 Parameter Assignment

The following sections explain how you can assign parameters for data transmission with the "open driver" in mode 1 and how you should structure the required data blocks.

You have two options:

- either assign parameters directly in STEP 5 by inputting them into data blocks
- or enter the parameters via the menus in the COM PP Parameter Assignment Software and then transfer the data blocks you have created to the CPU using the STEP 5 basic package. Working with COM PP is described in Chapter 3.

Assigning parameters to DX 2

DX 2 contains the link type. It also contains the pointers to the required parameters and parameter sets. The following table shows the values you can use when assigning parameters to DX 2. The assignment in DX 2 begins at DW 0 (absolute). All the numerical values are in hex.

DW	Parameter	Significance
0	4D41	MA
1	534B	SK
2	5832	X2
Link type		
3	0030	Data transmission with the "open driver"
Pointer to static parameter set		
4	44xx or 58xx	DB no. xx or[4] DX no. xx
5	xxxx	from DW no. xxxx
-		
6	0000	reserved
7	0000	reserved
Pointer to transmit mailbox [1]		
8	44xx or 58xx	DB no. xx or [4] DX no. xx
9	xxxx	from DW no. xxxx
10	xxxx	Length of transmit mailbox in words (range of values 0002H to 0801H)
Pointer to receive mailbox [2]		
11	44xx or 58xx	DB no. xx or [4] DX no. xx
12	xxxx	from DW no. xxxx
13	xxxx	Length of receive mailbox in words (range of values 0002H to 0801H)
Pointer to SCB/RCB [3]		
14	4D00 or 5300 or 44xx or 58xx	Flag or [4] S flag or DB no. xx or DX no. xx
15	xxxx	FW no., SW no. or DW no. xxxx

1) These parameters are not necessary if the CPU 928B or the CPU 948 is purely passive. In this case, enter the value 0000 in data words DW 8 to DW 10.
2) These parameters are not necessary if the CPU 928B or the CPU 948 is purely active, i.e. does not receive data. If this is the case, enter the value 0000 in data words DW 11 to DW 13.
3) The locations of SCB and RCB must be selected together. For a detailed description, see also Section 6.2.1.
4) Range of values for xx from 3 to FF (3 - 255)

Example

In this example you can see how to assign the following basic parameters to a DX 2:

- The static parameter set is in data block 10 from data word 4 onwards.
- The transmit mailbox is in data block 6 from data word 2 onwards, length 10 words.
- The receive mailbox is in data block 8 from data word 1 onwards, length 6 words.
- The send and receive coordination bytes (SCB and RCB) are in flag word 6.

With these parameters, DX 2 appears as below:

DX 2

0 :	KH = 4D41;	'MASKX2'
1 :	KH = 534B;	
2 :	KH = 5832;	
3 :	KH = 0030;	**LINK TYPE** "open driver"
4 :	KH = 440A;	**STATIC PARAMETER SET** IN DB 10
5 :	KH = 0004;	FROM DATA WORD 4
6 :	KH = 0000;	reserved
7 :	KH = 0000;	
8 :	KH = 4406;	**TRANSMIT MAILBOX** IN DATA BLOCK 6
9 :	KH = 0002;	FROM DATA WORD 2
10:	KH = 000A;	LENGTH 10 WORDS
11:	KH = 4408;	**RECEIVE MAILBOX** IN DATA BLOCK 8
12:	KH = 0001;	FROM DATA WORD 1
13:	KH = 0006;	LENGTH 6 WORDS
14:	KH = 4D00;	**COORDINATION BYTES** IN FLAG AREA
15:	KH = 0006;	SCB FLAG BYTE 6, RCB FLAG BYTE 7
16:	KH = xxxx;	no longer belongs to the parameter assignment for DX 2, content irrelevant, does not need to exist
:		
:		

Static parameter set

In the static parameter set, you must enter the parameters for the physical layer as well as specific transmission parameters.

The following table shows the values that you can enter in a data block or extended data block for the static parameter set. The parameter assignment begins at the DW specified in DX 2.

All numerical values are shown in hexadecimal format.

DW (rel.)	Range of values	Significance
Transmission speed		
0	0001 0002 0003 0004 0005 0006 0007 0008 0009	110 bps 150 bps 300 bps 600 bps 1200 bps 2400 bps 4800 bps 9600 bps 19200 bps (only with V.24 submodule and RS 422-A/485 submodule)
Parity		
1	0000 0001 0002	no odd even
Bits per character		
2	0006 0007 0008	6 bits per character 7 bits per character 8 bits per character
Stop bits		
3	0001 0002 0003	1 stop bit 1 + 1/2 stop bits 2 stop bits
Flow control		
4	0000 0001 0002	none XON/XOFF RTS/CTS
Mode		
5	0001	fixed user data
-		
6	0000	reserved
Character delay time		
7	0001 to FFFF	monitoring time 0.01 sec. to 655.35 sec. (unit: 10 ms)
8	0000	reserved
9	0000	reserved
10	0000	reserved

Caution

Remember that data words 8 to 10 must nevertheless exist.

Note

To avoid corrupting frames, it is advisable to operate with flow control.

Meaning of the parameters

Transmission speed	Specified in bps.
Parity	Number of bits with the value "1" in a string of information bits with a uniform length (e.g. 1 byte). Depending on the value, the parity is "even" or "odd". To check the parity, the string of information bits is extended by a further bit, the parity bit ("0" or "1") which when added to the information bits produces a selectable parity state.
Bits per character	Number of data bits used to form a character.
Stop bits	Duration of the stop bits, relative to the time required to transmit an information bit. The stop bits follow each character transmitted in a start-stop transmission.
Flow control	A mechanism with which the transmission (CPU or partner) and reception (partner or CPU) of data is synchronized when the data source operates faster than the data sink. The control of the data flow using flow control is transparent for the user (with the exception of the required parameter assignment). This means that if you use flow control 0001 (XON/XOFF, control characters DC1 and DC3), the user data must not contain these control characters. If you use flow control 0002 (RTS/CTS) you must also wire signals RTS and CTS (refer to Section 2.4.3 "Wiring a Connecting Cable for RTS/CTS Flow Control").

Mode

Version of the possible methods of data transmission

Mode 1 (fixed user data length)

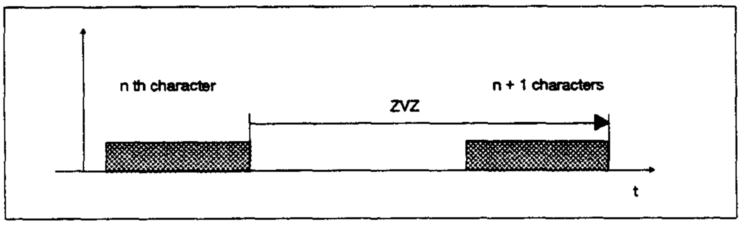
Sending When sending n bytes of user data, you must specify the number n of user data to be transmitted in the first word of the transmit mailbox. This specified length does not include the word containing the length specification itself.

Receiving When receiving n bytes of user data, the system program enters the number n of received useful bytes in the first word of the receive mailbox. This length does not include the word containing the length specification.

The end of a message is detected when the character delay time has elapsed.

Character delay time (ZVZ)

The maximum permitted time interval between two received characters (see following table).



Transmission speed	Min. permitted ZVZ
110 bps	120 ms
150 bps	90 ms
300 bps	50 ms
600 bps	30 ms
1200 bps	20 ms
2400 bps	10 ms
4800 bps	10 ms
9600 bps	10 ms
19200 bps	10 ms

Example

The first example indicates how to write a static parameter set in DB 10 from DW 4 for a point-to-point link in transmission mode 1.

DB 10

0:	KH = xxxx;	does not belong to the static parameter set,
1:	KH = xxxx;	content irrelevant
2:	KH = xxxx;	
3:	KH = xxxx;	
4:	KH = 0007;	**transmission speed** = 4800 bps
5:	KH = 0001;	**parity** = 1 = odd
6:	KH = 0006;	**bits per character** = 6
7:	KH = 0002;	**stop bits** = 1 + 1/2
8:	KH = 0000;	**no flow control**
9:	KH = 0002;	**mode** = 1 = variable user data length
10:	KH = 0000;	reserved
11:	KH = 0037;	**character delay time** = 55 x 0.01sec (=550 ms)
12:	KH = 0000;	
13:	KH = 0000;	reserved
14:	KH = 0000;	
15:	KH = yyyy;	no longer required in mode 1, content irrelevant,
:		does not need to exist
:		

6

Transmit mailbox

The transmit mailbox is located in a data block (DB) or extended data block (DX).

The transmit mailbox contains the data to be sent as well as the length specification in the first word (in bytes). This means that you can decide how many bytes of data are to be sent to the partner for each job. The specified length does not include the word containing the length specification itself. The length specification is not transmitted.

The length of the data to be transmitted must not exceed the length of the transmit mailbox minus one word (see above). The length of the transmit mailbox is a maximum of 2049 words (4098 bytes). You specify this in DX 2.

Remember to specify the length of the transmit mailbox in **words** in DX2, but the length of the data to be transmitted in **bytes** in the first word of the transmit mailbox. Select the length of the transmit mailbox as suitable for your particular application.

Example: transmit mailbox

The example illustrates how to assign parameters for a transmit mailbox in data block 6 from data word 2 onwards with a length of 10 words. You want to transmit 9 bytes.

DB 6

0:	KH = xxxx;	does not belong to transmit mailbox, content irrelevant
1:	KH = xxxx;	
2:	KH = 0009;	start of transmit mailbox, length in bytes
3:	KH = 0102;	data to be transmitted: 0102
4:	KH = 0304;	data to be transmitted: 0304
5:	KH = 0506;	data to be transmitted: 0506
6:	KH = 0708;	data to be transmitted: 0708
7:	KH = 09yy;	data to be transmitted: 09, yy is not transmitted
8:	KH = yyyy;	the content of the transmit mailbox is no longer sent
9:	KH = yyyy;	from this point
10:	KH = yyyy;	
11:	KH = yyyy;	
12:	KH = xxxx;	no longer belongs to the transmit mailbox, content irrelevant, does not need to exist
:		
:		

Note

For data transmission with the "open driver" the parameters of the physical layer on the communications partner must match those of the CPU 928B/CPU 948.

Receive mailbox

The receive mailbox is located in a data block (DB) or extended data block (DX).

The receive mailbox contains the received data and the length of the received data specified in the first word (in bytes). This is entered by the system program so that you can recognize how many bytes have been received. The specified length does not include the word containing the length specification itself.

The length of the receive mailbox is a maximum of 2049 words (4098 bytes). You specify this length (in words) in DX 2. You can only receive as many bytes as you specify in DX 2 (in words) for the length of the receive mailbox minus the first word containing the length specification in the receive mailbox.

Example: receive mailbox

The example illustrates the structure of a receive mailbox in data block 8 from data word 1 onwards with a length of 6 words. Seven bytes were received.

DB 8

0:	KH = xxxx;	does not belong to receive mailbox, content irrelevant
1:	KH = 0007;	start of the receive mailbox, length specified in bytes
2:	KH = 0102;	received data: 0102
3:	KH = 0304;	received data: 0304
4:	KH = 0506;	received data: 0506
5:	KH = 07yy;	received data: 07, yy was not received
6:	KH = yyyy;	content of the receive mailbox, not received
7:	KH = xxxx;	does not belong to the receive mailbox, content irrelevant, does not need to exist
:		
:		

Send and receive coordination bytes (SCB and RCB)

With the SCB you can start the following jobs:

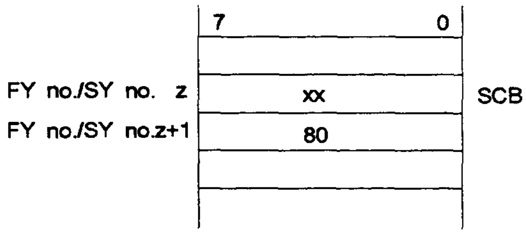
- SEND job
- reset the "open driver"

With the RCB you can perform the following

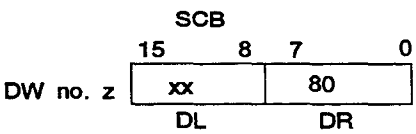
- enable and disable the receive mailbox

The coordination bytes can be flags or S flags or be located in a data block or extended data block. Flags and S flags already exist on the CPU 928B or on the CPU 948, you must set up data blocks and extended data blocks yourself.

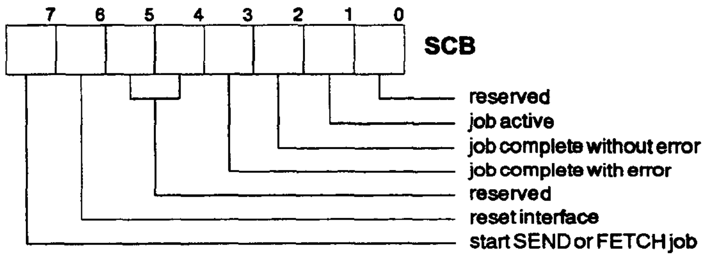
Representation in the flag or S flag area



Representation in a data block or extended data block



Send coordination byte (SCB) The send coordination byte is structured as follows:



The individual bits or bit groups, providing they are not reserved, are described below.

Bit 1 "Job active"

- set by the system program when a job starts
- reset by the system program when a job is completed
- you can read this bit, to know whether there is any point in starting a new job

Bit 2 "Job complete without error"

- set by the system program when a job is completed without error
- you can read this bit and then reset it, or it is reset by the system program when bit 1 "job active" is set for a new job
- you can read this bit, to know whether a job was completed without error

Bit 3 "Job terminated with error"

- set by the system program when a job is terminated with an error
- you can read this bit and then reset it, or it is reset by the system program when bit 1 "job active" is set for a new job
- you can read this bit, to know whether a job was completed with an error

Bit 6 "Reset interface"

- you can set this bit to reset the interface
- reset by the system program as soon as the system program resets the interface

Bit 7 "Start SEND job"

- you can set this bit to start a SEND job
- reset by the system program as soon as the system program completes the SEND job

When you reset the interface, the parameters in the static parameter set (from DW 5 onwards) are read in again:

- mode
- character delay time

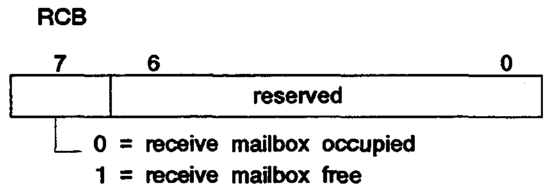
If the SCB bits 6 and 7 are sent simultaneously or if SCB bit 6 is set first followed by SCB bit 7 at an interval ≤ 10 ms, the system program first carries out the reset followed by the SEND job. If SCB bit 7 is set first followed by SCB bit 6 at an interval ≤ 10 ms, the sequence of the two jobs cannot be guaranteed. By resetting, you abort the currently active partner SEND job.

Note

You can only modify the parameters of the physical layer involving the physical transmission characteristics and the flow control (static parameter set up to and including DW 4) using a cold restart on the CPU 928B or on the CPU 948 using a cold restart or power up. Resetting the interface is not sufficient.

Receive coordination byte (RCB)

As long as the RCB bit 7 is reset, no received data are transferred to the receive mailbox.



Caution

In your user program, you should make sure that the received data are processed as quickly as possible and that the receive mailbox is enabled immediately afterwards by RCB bit 7.

Caution

During a cold restart and following power up, 00H is written to the SCB (no job active) and 80H to the RCB (receive mailbox free).

6.2.2 Getting Started

Preparations

You have already installed the interface submodule in the CPU 928B or in the CPU 948 (see Chapter 2), the CPU is plugged into the PLC rack and the power supply is off.

Procedure

We recommend the following procedure for starting data transmission with the "open driver":

Step	Action
1	Connect the CPU 982B/CPU 948 to the partner (with a standard connecting cable).
2	Switch on the power supply.
3	Transfer your user program to the CPU 928B. The user program must contain the following **data blocks** or **extended data blocks** to be able to set the SI 2 interface for data transmission with the "open driver". DX 2 DB/DX with the static parameter set DB/DX with the transmit mailbox [1] DB/DX with the receive mailbox [2] DB/DX with the SCB and RCB (only when the SCB/RCB are located in a DB/DX [3]
4	Perform a **cold restart**.
5	Prepare the partner for data transmission with the "open driver".

[1] Only when the CPU 928B/CPU 948 triggers active jobs itself
[2] Only when the CPU 928B/CPU 948 receives passive jobs
[3] If the SCB and RCB are flags or S flag words, they already exist in the CPU 928B/ CPU 948 and do not need to be transferred.

6

CPU 928B

During a cold restart or following power up the system program checks that the following exist:

- DX 2,
- the static parameter set
- SCB and RCB

and the correct parameter assignment in DX 2

DX 2 exists and has correct parameters

- The system program initializes the SI 2 interface for data transmission with the "open driver".
- The red LED SI 2 goes off as soon as data transmission with the "open driver" is ready for operation.
- The system program resets the SCB bit 7 (i.e. no SEND job active).
- The system program sets the SCB bit 7 (i.e. receive mailbox enabled).

↓

Data transmission with the "open driver" is ready for operation

DX 2 does not exist or does not have correct parameters

- The CPU remains in the STOP mode.
- The red LED SI 2 is lit in addition to the STOP LED.
- The system program outputs an error message to RS 3 and RS 4.

CPU 948

During a cold restart or following power up the system program checks that the following exist:

- DX 2
- the static parameter set
- SCB and RCB

and the correct parameter assignment in DX 2

DX 2 exists and has correct parameters

- The system program initializes the SI 2 interface for data transmission with the "open driver".
- The red LED SI 2 goes off as soon as data transmission with the "open driver" is ready for operation.
- The system program resets the SCB bit 7 (i.e. no SEND job active).
- The system program resets the RCB bit 7 (i.e. receive mailbox enabled).

↓

Data transmission with the "open driver" is ready for operation

DX 2 does not exist or does not have correct parameters

- The system program does not initialize the SI 2 interface for data transmission with the "open driver"
- The red LED SI 2 does not go off

6.2.3 Operation

Preparations

The SI 2 interface has parameters assigned for data transmission with the "open driver". The SI 2 LED is off. Data transmission with the "open driver" is ready for operation.

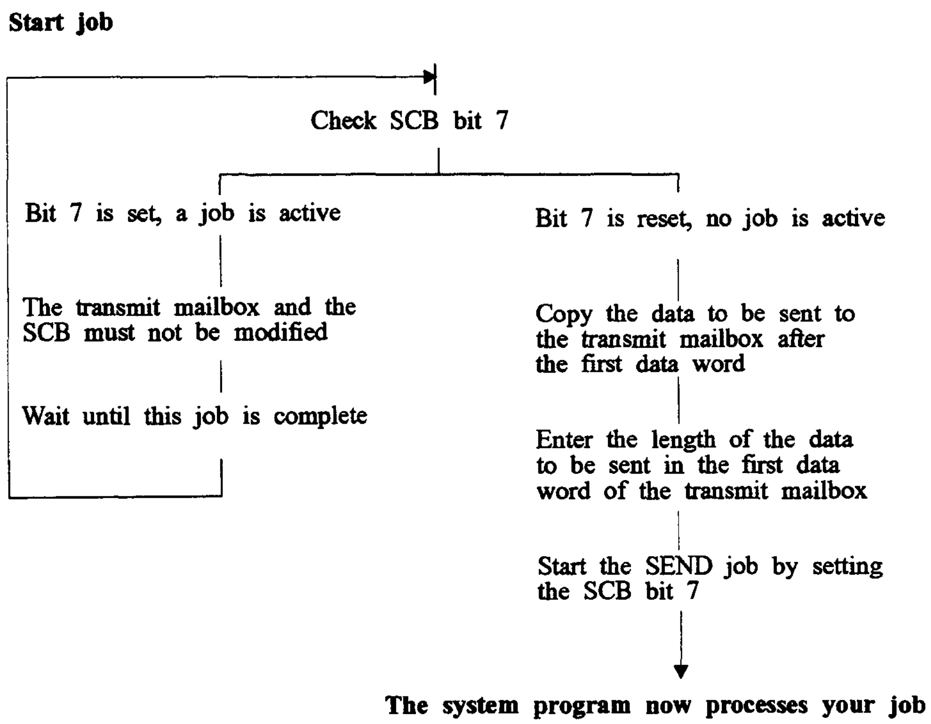
Method

The system program now checks constantly whether the send coordination byte (SCB) and receive coordination byte (RCB) exist. If they do not exist (e.g. because the DB/DX has been deleted), the system program calls OB 35. The error number is entered in ACCU 1 (see Chapter 11).

The system program now processes the jobs you have programmed in your user program, or the jobs triggered by the communications partner. The following sections explain how to program the jobs in your user program.

SEND job

If you want to program a SEND job, remember the following sequence for the various modes:



The following diagram illustrates the reactions that can occur if the SCB bit 7 is set and the system program is processing your job.

SCB bit 7 is set

The transmit mailbox exists

The system program transfers the data specified by the length word (first word in the transmit mailbox). The length word is not transmitted.

The job was completed without errors

The system program

- transferred the data from the transmit mailbox to the partner
- reset the SCB bit 7

The job was completed with an error

The system program

- called OB 35; ACCU 1 contains the error number
- reset the SCB bit 7

The transmit mailbox does not exist

The system program

- did not process the job
- called OB 35 and entered an error number in ACCU 1
- reset the SCB bit 7

Note

You can only start a new SEND job when the previous job is completed.

If a SEND job was completed with an error, you must wait for approximately twice the character delay time of the communications partner before you can start a new SEND job, otherwise further errors will result at the communications partner.

Resetting the interface

You have set the SCB bit 6 and have therefore started a reset job. The following diagram shows the possible reactions.

SCB bit 6 is set

The CPU 928B/CPU 948 is currently processing a job from the partner

- the partner's job is aborted
- OB 35 has been called: ACCU 1 contains the error number of the aborted job on the partner
- the transmit and receive buffers are cleared (see note)
- the static parameters from DW 5 onwards are read and evaluated
- when the reset job is complete, the SCB bit 6 is reset

The CPU 928B/CPU 948 is not processing a job from the partner

- the static parameters from DW 5 onwards are read in and evaluated
- when the reset job is complete, the SCB bit 6 is reset

> **Note**
>
> A reset job can mean termination in the middle of the protocol. To avoid errors resulting from this, you should wait for approximately twice the character delay time of the communications partner before starting the next SEND job.

SEND job on the partner

SEND jobs on the partner are not processed by the user program, but automatically by the system program in the background. The system program resets the RCB bit 7 (receive mailbox disabled) when the data have been entered completely in the receive mailbox. You can now evaluate the received data. The receive mailbox can only be overwritten when you have enabled it by setting RCB bit 7.

> **Caution**
>
> In your user program, make sure that the received data are processed as quickly as possible and then enable the receive mailbox immediately by setting the RCB bit 7.

CPU 928B

The following diagram illustrates the situations that can occur while a partner SEND job is being executed in mode 1.

Job from partner active

The CPU 928B is not in the STOP mode or wait state (program test)

The receive mailbox is disabled, RCB bit 7 is reset

The receive mailbox is enabled, RCB bit 7 is set

The job was completed without errors

The system program
- transferred the partner's data to the receive mailbox after the first data word
- entered the number of received data (specified in bytes) in the first data word of the receive mailbox
- reset the RCB bit 7

The job was completed with an error

The system program
- aborted the partner's SEND job with an error
- called OB 35; ACCU 1 contains the error number

The CPU 928B is in the STOP mode or wait state (program test)

The system program
- aborted the partner's SEND job with an error
- called OB 35 after exiting the STOP or wait mode; ACCU 1 contains the error number

CPU 948

The following diagram illustrates the situations that can occur while a partner SEND job is being executed in mode 1.

Job from partner active

The receive mailbox is disabled,
RCB bit 7 is reset

The receive mailbox is enabled,
RCB bit 7 is set

The job was completed without errors

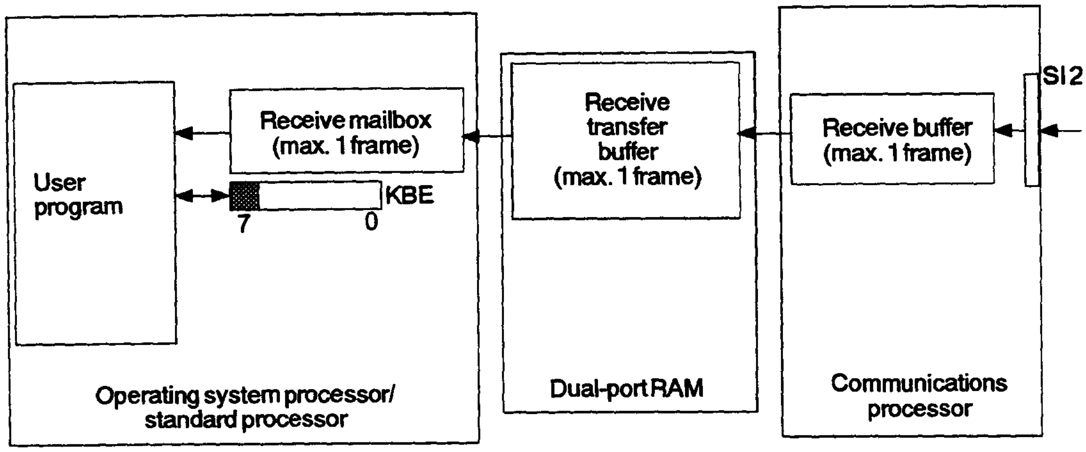
The system program
- transferred the partner's data to the receive mailbox after the first data word
- entered the number of received data (specified in bytes) in the first data word of the receive mailbox
- reset the RCB bit 7

The job was completed with an error

The system program
- aborted the partner's SEND job with an error
- called OB 35: ACCU 1 contains the error number

Points to note when receiving

Only one message can ever be entered in the receive mailbox (max. 4096 bytes long), although the receive transfer buffer of the dual-port RAM and the receive buffer can both hold one further message. Once you empty the mailbox and set bit 7 (receive mailbox free) in the receive coordination byte (RCB), the next message, if it exists, is transferred from the receive transfer buffer of the dual-port RAM into the mailbox and the next message shifted from the receive buffer to the receive transfer buffer of the dual-port RAM.



Response to special operating modes

Stop mode

CPU 928B

In the "STOP" mode, user programs are not processed, i.e. you cannot start a new job. When the CPU changes from "RUN" to "STOP", the following applies to an already active job started by the user program:

- The job is stopped with an error, if there is still data to be transferred between the operating system processor and the dual-port RAM.

- The job is completed normally if there is no more data to be transferred between the operating system processor and the dual-port RAM.

If the partner triggers a job in the "STOP" mode, this is acknowledged negatively. An appropriate error message is entered in the error area of the interface (refer to Chapter 11 "Error Messages"). When the CPU changes from the "RUN" mode to the "STOP" mode, an already active job started by the partner is stopped with an error, i.e. negatively acknowledged. An appropriate error message is entered in the error area of the interface (refer to Chapter 11 "Error Messages")

CPU 948

In contrast to the CPU 928B, the CPU 948 also supports communication in "soft" stop and in the wait mode (program test).

Wait mode

CPU 928B

The CPU 928B goes into the wait mode when the PG online function "PROGRAM TEST" is active. In the wait mode user programs are not processed, i.e. you cannot start a new job. An already active job started by the user is continued and completed after the wait mode is exited.

If the partner triggers a job in the wait mode, this is acknowledged negatively. An appropriate error message is entered in the error area of the interface (refer to Chapter 11 "Error Messages"). When the CPU changes to the wait mode, an already active job started by the partner is negatively acknowledged. An appropriate error message is entered in the error area of the interface (refer to Chapter 11 "Error Messages")

CPU 948

In contrast to the CPU 928B, the CPU 948 also supports communication in "soft" stop and in the wait mode (program test).

6.2.4 Protocol Definition of the "Open Driver" in Mode 1

The following section is intended to help you if you want to connect a non-SIMATIC device using the "open driver" in mode 1 (fixed user data length). It is also intended, however, as a source of reference for engineers who have to develop a protocol compatible with the "open driver" on a non-Simatic computer.

The following aspects of transmitting and receiving are described:

- structure of the transmitted/received data frame
- any restrictions
- basic sequence of events

The sequence of events refer to the communications processor. The heavy lines indicate the standard sequence, whereas the thinner lines represent error handling.

Transmitting

Setting up the transmit mailbox is described in Section 6.2.1 "Transmit mailbox". The sequence of the SEND job is described in Section 6.2.3 "SEND job".

In mode 1, only the user data are transmitted. The length information contained in the first word of the transmit mailbox is not transmitted.

The data frame to be transmitted is structured as shown below:

Length of the user data: N bytes

1	N
User data - length N bytes	

Caution
If you select flow control with XON/XOFF, the user data must not contain XON (DC1 = 11H) or XOFF (DC3 = 13H).

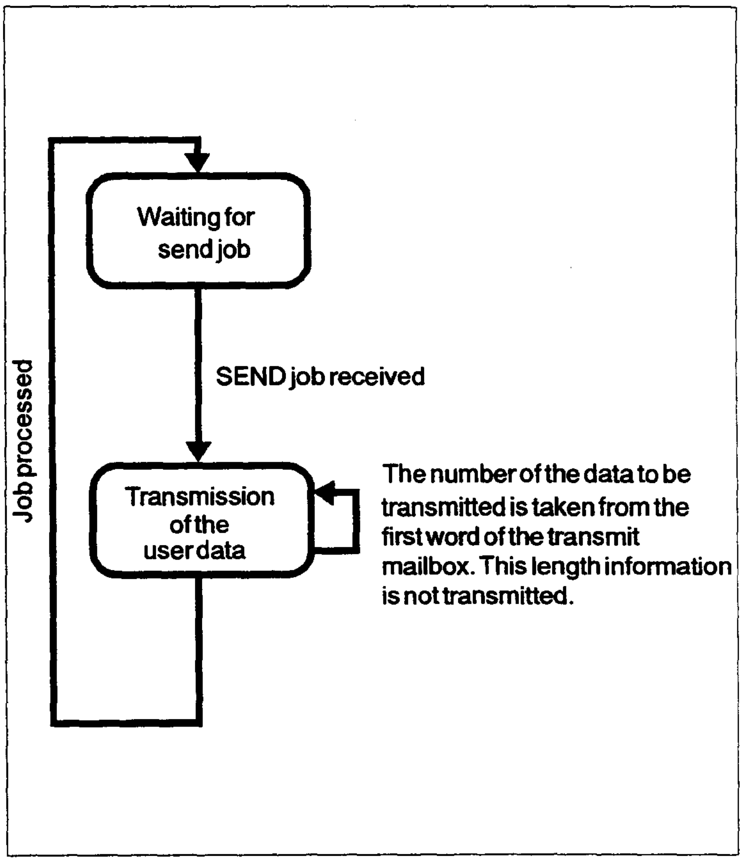


Fig. 6-6 Sequence of transmission in mode 1

Receiving

The setting up of the receive mailbox is described in Section 6.2.1 "Receive mailbox". The sequence of the partner SEND job is described in Section 6.2.3 "Send job on the partner".

The end of the frame is recognized when the character delay time elapses.

The data frame to be received has the following structure:

Length of the user data: N bytes

1 ... N
User data - length N bytes

When the character delay time (ZVZ) elapses, this marks the valid end of the frame.

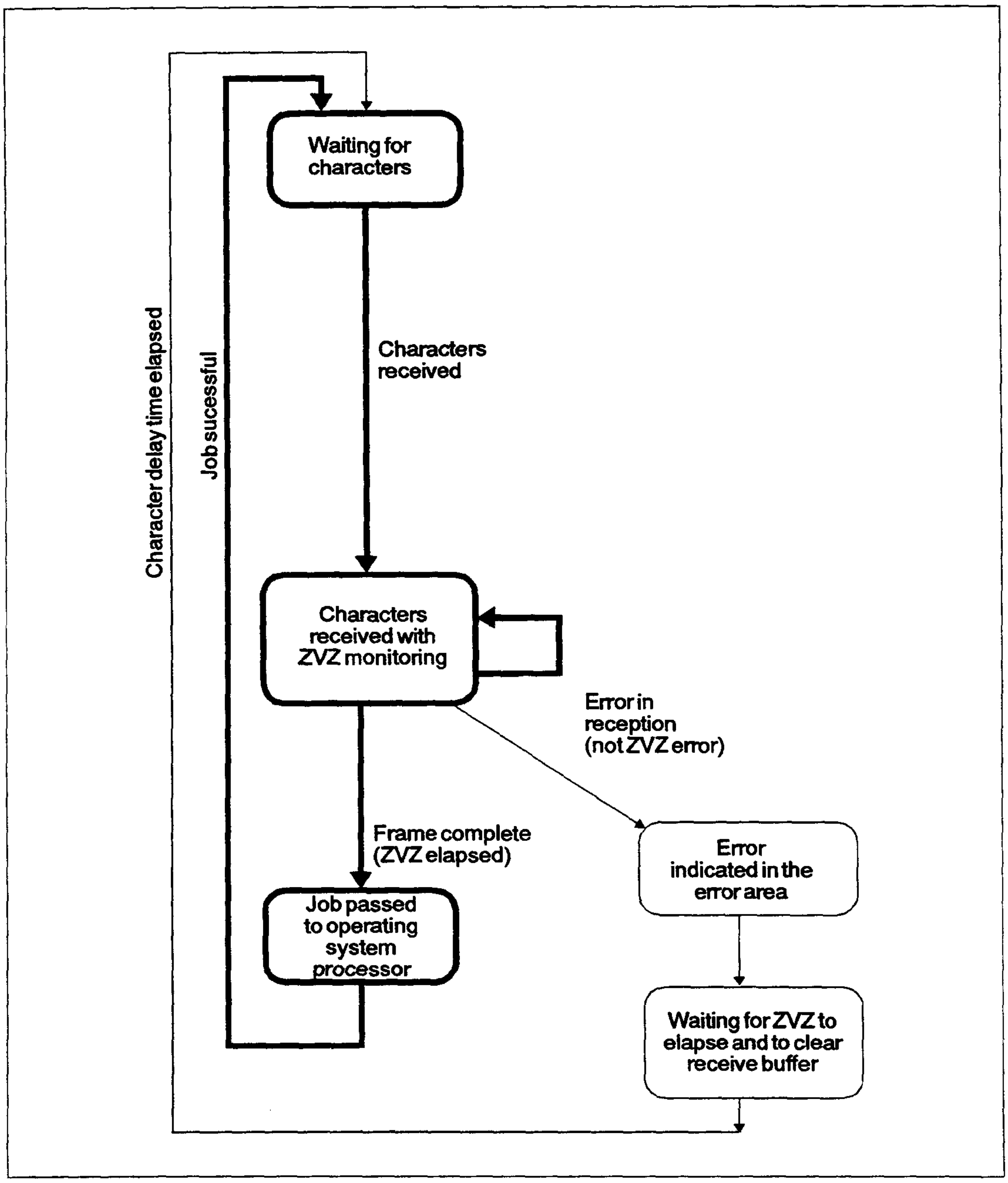


Fig. 6-7 Sequence of receiving in mode 1

6.2.5 Processing Times

This section lists the typical processing times for a SEND job.

Processing times for the CPU 928B

The processing times are subject to the following conditions:

- two CPU 928Bs are connected
- both the active and passive CPU have parameters assigned for data transfer with the "open driver" in mode 1 (fixed user data length). In the static parameter set a character delay time of 220 ms is selected (even parity, 8 bits per character, 1 stop bit).

The processing time is the time
- from starting the job on the active CPU 928B (the user program sets SCB bit 7)
- until the job is completed on the active CPU 928B (the system program resets SCB bit 7)

SEND job

The times were measured
- with different transmission speeds and
- with various amounts of user data

No. of bytes	Transmission speed								
	110	150	300	600	1200	2400	4800	9600	19200
1	0.336s	0.296s	0.276s	0.226s	0.266s	0.266s	0.266s	0.266s	0.257s
2	0.339s	0.299s	0.279s	0.269s	0.269s	0.269s	0.269s	0.269s	0.259s
4	0.632s	0.519s	0.388s	0.319s	0.289s	0.279s	0.269s	0.269s	0.259s
8	1.032s	0.808s	0.529s	0.390s	0.329s	0.299s	0.279s	0.274s	0.259s
16	1.832s	1.393s	0.826s	0.539s	0.399s	0.329s	0.299s	0.279s	0.268s
32	2.432s	2.562s	1.409s	0.829s	0.549s	0.409s	0.339s	0.299s	0.270s
64	6.632s	4.910s	2.579s	1.419s	0.839s	0.558s	0.409s	0.339s	0.289s
128	13.031s	9.608s	4.929s	2.592s	1.429s	0.849s	0.559s	0.409s	0.329s
256	25.829s	18.922s	9.625s	4.939s	2.599s	1.436s	0.849s	0.559s	0.399s
384	38.634s	28.377s	14.319s	7.289s	3.779s	2.019s	1.139s	0.709s	0.479s
512	51.336s	37.696s	18.990s	9.631s	4.949s	2.609s	1.439s	0.849s	0.549s
1024	102.640s	75.310s	37.780s	19.020s	9.639s	4.957s	2.609s	1.439s	0.839s
2048	205.032s	150.400s	75.328s	37.790s	19.028s	9.649s	4.959s	2.609s	1.429s
3072	307.456s	225.480s	112.864s	56.566s	28.419s	14.339s	7.299s	3.789s	2.019s
4096	409.856s	300.560s	150.408s	75.332s	37.800s	19.036s	9.648s	4.959s	2.609s

Processing times for the CPU 948

The processing times are subject to the following conditions:

- two CPU 948s are connected
- both the active and passive CPU have parameters assigned for data transfer with the "open driver" in mode 1 (fixed user data length). In the static parameter set a character delay time of 220 ms is selected (even parity, 8 bits per character, 1 stop bit).

The processing time is the time
- from starting the job on the active CPU 948 (the user program sets SCB bit 7)
- until the job is completed on the active CPU 948 (the system program resets SCB bit 7)

SEND job

The times were measured
- with different transmission speeds and
- with various amounts of user data

No. of bytes	Transmission speed								
	110	150	300	600	1200	2400	4800	9600	19200
1	0.487s	0.427s	0.346s	0.307s	0.287s	0.277s	0.277s	0.279s	0.279s
2	0.582s	0.499s	0.379s	0.323s	0.299s	0.279s	0.279s	0.279s	0.279s
4	0.782s	0.639s	0.452s	0.359s	0.309s	0.289s	0.279s	0.279s	0.279s
8	1.182s	0.939s	0.599s	0.436s	0.349s	0.309s	0.289s	0.285s	0.279s
16	1.982s	1.519s	0.897s	0.578s	0.420s	0.349s	0.309s	0.289s	0.289s
32	2.582s	2.699s	1.479s	0.875s	0.569s	0.419s	0.349s	0.309s	0.295s
64	6.783s	5.039s	2.657s	1.459s	0.859s	0.569s	0.419s	0.349s	0.309s
128	13.183s	9.739s	4.999s	2.634s	1.449s	0.859s	0.569s	0.419s	0.349s
256	25.985s	19.121s	9.698s	4.979s	2.624s	1.449s	0.859s	0.569s	0.419s
384	38.787s	28.510s	14.389s	7.329s	3.799s	2.029s	1.159s	0.719s	0.499s
512	51.589s	37.899s	18.080s	9.679s	4.969s	2.619s	1.449s	0.859s	0.569s
1024	102.797s	75.449s	37.859s	19.061s	9.669s	4.969s	2.619s	1.449s	0.869s
2048	205.212s	150.548s	75.410s	37.839s	19.059s	9.659s	4.969s	2.629s	1.458s
3072	307.627s	225.643s	112.960s	56.619s	28.449s	14.359s	7.319s	3.799s	2.040s
4096	410.043s	300.741s	150.510s	75.390s	37.839s	19.059s	9.669s	4.979s	2.629s

6.3 Mode 2 - Variable User Data Length

6.3.1 Parameter Assignment

The following sections explain how you can assign parameters for data transmission with the "open driver" in mode 2 and how you should structure the required data blocks.

You have two options:

- either assign parameters directly in STEP 5 by inputting them into data blocks
- or enter the parameters via the menus in the COM PP Parameter Assignment Software and then transfer the data blocks you have created to the CPU using the STEP 5 basic package. Working with COM PP is described in Chapter 3.

Assigning parameters to DX 2

DX 2 contains the link type. It also contains the pointers to the required parameters and parameter sets. The following table shows the values you can use when assigning parameters to DX 2. The assignment in DX 2 begins at DW 0 (absolute). All the numerical values are in hex.

DW	Parameter	Significance
0	4D41	MA
1	534B	SK
2	5832	X2
Link type		
3	0030	Data transmission with the "open driver"
Pointer to static parameter set		
4	44xx or 58xx	DB no. xx or [4] DX no. xx
5	xxxx	from DW no. xxxx
-		
6	0000	reserved
7	0000	reserved
Pointer to transmit mailbox [1]		
8	44xx or 58xx	DB no. xx or [4] DX no. xx
9	xxxx	from DW no. xxxx
10	xxxx	Length of transmit mailbox in words (range of values 0002H to 0801H)
Pointer to receive mailbox [2]		
11	44xx or 58xx	DB no. xx or [4] DX no. xx
12	xxxx	from DW no. xxxx
13	xxxx	Length of receive mailbox in words (range of values 0002H to 0801H)
Pointer to SCB/RCB [3]		
14	4D00 or 5300 or 44xx or 58xx	Flag or [4] S flag or DB no. xx or DX no. xx
15	xxxx	FW no., SW no. or DW no. xxxx

1) These parameters are not necessary if the CPU 928B/CPU 948 is purely passive. In this case, enter the value 0000 in data words DW 8 to DW 10.

2) These parameters are not necessary if the CPU 928B/CPU 948 is purely active, i.e. does not receive data. If this is the case, enter the value 0000 in data words DW 11 to DW 13.

3) The locations of SCB and RCB must be selected together. For a detailed description, refer to Section 6.3.1.

4) Range of values for xx from 3 to FF (3 - 255)

Example

In this example you can see how to assign the following basic parameters to a DX 2:

- The static parameter set is in data block 10 from data word 4 onwards.
- The transmit mailbox is in data block 6 from data word 2 onwards, length 10 words.
- The receive mailbox is in data block 8 from data word 1 onwards, length 6 words.
- The send and receive coordination bytes (SCB and RCB) are in flag word 6.

With these parameters, DX 2 appears as below:

DX 2

0 :	KH = 4D41;	
1 :	KH = 534B;	'MASKX2'
2 :	KH = 5832;	
3 :	KH = 0030;	**LINK TYPE** "open driver"
4 :	KH = 440A;	**STATIC PARAMETER SET** IN DB 10
5 :	KH = 0004;	FROM DATA WORD 4
6 :	KH = 0000;	reserved
7 :	KH = 0000;	
8 :	KH = 4406;	**TRANSMIT MAILBOX** IN DATA BLOCK 6
9 :	KH = 0002;	FROM DATA WORD 2
10:	KH = 000A;	LENGTH 10 WORDS
11:	KH = 4408;	**RECEIVE MAILBOX** IN DATA BLOCK 8
12:	KH = 0001;	FROM DATA WORD 1
13:	KH = 0006;	LENGTH 6 WORDS
14:	KH = 4D00;	**COORDINATION BYTES** IN FLAG AREA
15:	KH = 0006;	SCB FLAG BYTE 6, RCB FLAG BYTE 7
16:	KH = xxxx;	no longer belongs to the parameter assignment for DX 2, content irrelevant, does not need to exist
:		
:		

Static parameter set

In the static parameter set, you must enter the parameters for the physical layer as well as specific transmission parameters.

The following table shows the values that you can enter in a data block or extended data block for the static parameter set. The parameter assignment begins at the DW specified in DX 2.

All numerical values are shown in hexadecimal format.

DW (rel.)	Range of values	Significance
Transmission speed		
0	0001 0002 0003 0004 0005 0006 0007 0008 0009	110 bps 150 bps 300 bps 600 bps 1200 bps 2400 bps 4800 bps 9600 bps 19200 bps (only with V.24 and RS 422-A/485 submodules)
Parity		
1	0000 0001 0002	no odd even
Bits per character		
2	0006 0007 0008	6 bits per character 7 bits per character 8 bits per character
Stop bits		
3	0001 0002 0003	1 stop bit 1 + 1/2 stop bits 2 stop bits
Flow control		
4	0000 0001 0002	none XON/XOFF RTS/CTS
Mode		
5	0002	variable user data length
-		
6	0000	reserved
Character delay time		
7	0001 to FFFF	monitoring time 0.01 sec. to 655.35 sec. (unit: 10 ms)
8	0000	reserved
9	0000	reserved
10	0000	reserved

DW (rel.)	Range of values	Significance
End identifier		
11	0000	00 (NUL) code of the first end identifier no second end identifier
	00xx	xx code of the second end identifier (range of values 01 - FF)
12	0000	no second end identifier
	00xx	xx code of the second end identifier (range of values 01 - FF)

Note
To avoid corrupting frames it is advisable to operate with flow control.

6

Meaning of the parameters

Transmission speed — Specified in bps.

Parity — Number of bits with the value "1" in a string of information bits with a uniform length (e.g. 1 byte). Depending on the value, the parity is "even" or "odd". To check the parity, the string of information bits is extended by a further bit, the parity bit ("0" or "1") which when added to the information bits produces a selectable parity state.

Bits per character — Number of data bits used to form a character.

Stop bits — Duration of the stop bits, relative to the time required to transmit an information bit. The stop bits follow each character transmitted in a start-stop transmission.

Flow control — A mechanism with which the transmission (CPU or partner) and reception (partner or CPU) of data is synchronized when the data source operates faster than the data sink.
The control of the data flow using flow control is transparent for the user (with the exception of the required parameter assignment). This means that if you use flow control 0001 (XON/XOFF, control characters DC1 and DC3), the user data must not contain these control characters.
If you use flow control 0002 (RTS/CTS) you must also wire signals RTS and CTS (refer to Section 2.4.3 "Wiring a Connecting Cable for RTS/CTS Flow Control").

Mode — Version of the possible methods of data transmission

Mode 2 (variable user data length)

Sending

When sending, the user data, up to and including the end identifier (as defined in relative DW 11 and relative DW 12 of the static parameter set) are transmitted from the transmit mailbox. You must specify the length of user data to be transmitted in the first word of the transmit mailbox. The end identifier must be part of the user data. This specified length does not include the word containing the length specification itself, which is not transmitted.

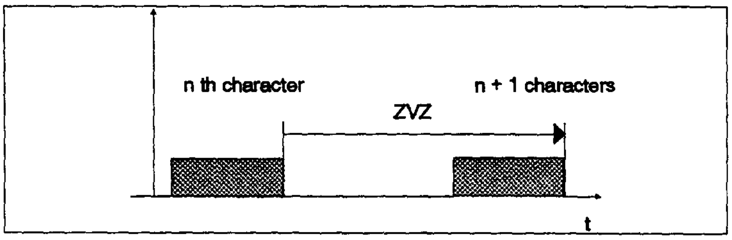
Receiving

When receiving, the user data, up to and including the end identifier (as defined in relative DW 11 and relative DW 12 of the static parameter set) are entered in the first word of the receive mailbox. The system program enters the number n of received user data (in bytes including the end identifier) in the first word of the recive mailbox. This length does not include the word containing the length specification.

If the character delay time elapses while receiving the data reception is terminated. The message is discarded.

Character delay time (ZVZ)

The maximum permitted time interval between two received characters (see following table).



Transmission speed	Min. permitted ZVZ
100 bps	120 ms
150 bps	90 ms
300 bps	50 ms
600 bps	30 ms
1200 bps	20 ms
2400 bps	10 ms
4800 bps	10 ms
9600 bps	10 ms
19200 bps	10 ms

End identifier

The data transmission is terminated by the specified end identifier. In the first end identifier, the entry NUL is a valid end character. This ensures that as standard, at least one valid end identifier is assigned. The second end identifier is optional and is only effective when the first end identifier is not NUL. In the second end identifier the entry NUL is counted as a non-assigned end character. If both end identifiers are specified (both < > NUL), then both together define the end of the message.

Caution

When assigning the end identifier(s) make sure that for data transmission with 6 or 7 bits per character, bit 6 or bit 7 of the assigned end identifier(s) are not transmitted.

Example

The example shows how to write a static parameter set in DB 10 from DW 4 for a point-to-point link in the transmission mode 2.

DB 10

0:	KH = xxxx;	does not belong to the static parameter set,
1:	KH = xxxx;	content irrelevant
2:	KH = xxxx;	
3:	KH = xxxx;	
4:	KH = 0007;	**transmission speed** = 4800 bps
5:	KH = 0001;	**parity** = 1 = odd
6:	KH = 0006;	**bits per character** = 6
7:	KH = 0002;	**stop bits** = 1 + 1/2
8:	KH = 0000;	**no flow control**
9:	KH = 0002;	**mode** = 2 = variable user data length
10:	KH = 0000;	reserved
11:	KH = 0037;	**character delay time** = 55 x 0.01sec (=550 ms)
12:	KH = 0000;	
13:	KH = 0000;	} reserved
14:	KH = 0000;	
15:	KH = 0010;	**code of the first end identifier** = 10H (DLE)
16:	KH = 0003;	**code of the second end identifier** = 03H (ETX)
17:	KH = yyyy;	no longer required in mode 2, content irrelevant,
:		does not need to exist
:		

Caution

In mode 2, make sure that the communications partner operates with the same end identifier.

Note

For data transmission with the "open driver" the parameters of the physical layer on the communications partner must match those of the CPU 928B or of the CPU 948.

Transmit mailbox

The transmit mailbox is located in a data block (DB) or extended data block (DX).

The transmit mailbox contains the data to be sent as well as the length specification in the first word (in bytes). The specified length does not include the word containing the length specification itself. Following this come the data to be transmitted and the end identifier selected in the static parameter set. The end identifier must be part of user data bytes specified in the first data word.

Make sure that the communications partner operates with the same end identifier as you selected in the static parameter set. The data in the transmit mailbox up to and including the assigned end identifier are transmitted. The length specified in the first data word must also be transmitted.

The length of the data to be transmitted must not exceed the length of the transmit mailbox minus one word (see above). The length of the transmit mailbox is a maximum of 2049 words (4098 bytes). You specify this in DX 2.

Remember to specify the length of the transmit mailbox in **words** in DX2, but the length of the data to be transmitted in **bytes** in the first word of the transmit mailbox. Select the length of the transmit mailbox as suitable for your particular application.

Example: transmit mailbox

The example illustrates how to assign parameters for a transmit mailbox in data block 6 from data word 2 onwards with a length of 10 words. The first end identifier 10H and the second end identifier 03H were selected in the static parameter set. The number of useful bytes relevant for transmission in this example is 12.

DB 6

0:	KH = xxxx;	does not belong to transmit mailbox, content irrelevant
1:	KH = xxxx;	
2:	KH = 000C;	start of transmit mailbox, relevant user data bytes:0CH
3:	KH = 0102;	data to be transmitted: 0102
4:	KH = 0304;	data to be transmitted: 0304
5:	KH = 0506;	data to be transmitted: 0506
6:	KH = 0710;	data to be transmitted: 07, first end identifier: 10
7:	KH = 03zz;	second end identifier: 03, zz is not transmitted
8:	KH = zzzz;	relevant user data, no longer sent here
9:	KH = yyyy;	the content of the transmit mailbox is no longer sent
10:	KH = yyyy;	from this point
11:	KH = yyyy;	
12:	KH = xxxx;	
:		does not belong to the transmit mailbox, content
:		irrelevant, does not need to exist

Receive mailbox

The receive mailbox is in a data block (DB) or extended data block (DX).

The receive mailbox contains the received data after the first data word including the received end identifier. The system program enters the number n (in bytes) of received user data bytes (including the end identifier) in the first word of the receive mailbox. This specified length does not include the word containing the length specification itself. Owing to the end identifier, you can determine how many bytes were received in addition to the first data word in the receive mailbox. Make sure that the communications partner operates using the same end identifier as you assigned in the static parameter set.

The length of the receive mailbox is a maximum of 2049 words (4098 bytes). You specify the length (in words) in DX 2. You can only receive as many bytes (including the end identifier) as you specified in DX 2 as the length (in words) of the receive mailbox, minus the first word in the receive mailbox containing the length specification.

Example: receive mailbox

The example illustrates the structure of a receive mailbox in data block 8 from data word 1 with a length of 6 words. In the static parameter set, you assigned 10H as the first end identifier and 03 as the second identifier.

DB 8

0:	KH = xxxx;	does not belong to receive mailbox, content irrelevant
1:	KH = 0009;	start of the receive mailbox, length specified in bytes
2:	KH = 0102;	received data: 0102
3:	KH = 0304;	received data: 0304
4:	KH = 0506;	received data: 0506
5:	KH = 0710;	received data: 07, first end identifier: 10
6:	KH = 03yy;	second end identifier: 03, yy was not received
7:	KH = xxxx;	does not belong to the receive mailbox, content irrelevant, does not need to exist
:		
:		

Send and receive coordination bytes (SCB and RCB)

With the SCB you can start the following jobs:

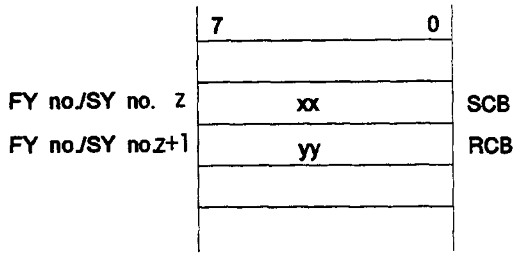
- **SEND jobs**
- **reset the "open driver"**

With the RCB you can perform the following

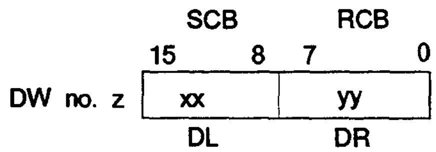
- **enable and disable the receive mailbox**

The coordination bytes can be flags or S flags or be located in a data block or extended data block. Flags and S flags already exist on the CPU 928B or on the CPU 948, you must set up data blocks and extended data blocks yourself.

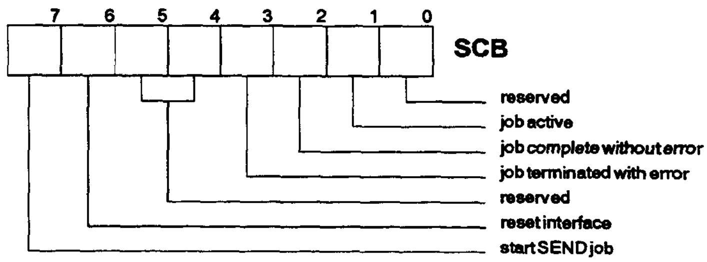
Representation in the flag or S flag area



Representation in a data block or extended data block



Send coordination byte (SCB) The send coordination byte is structured as follows:



The individual bits or bit groups, providing they are not reserved, are described below.

Bit 1 "Job active"

- set by the system program when a job starts
- reset by the system program when a job is completed
- you can read this bit, to know whether there is any point in starting a new job

Bit 2 "Job complete without error"

- set by the system program when a job is completed without error
- you can read this bit and then reset it, or it is reset by the system program when bit 1 "job active" is set for a new job

- you can read this bit, to know whether a job was completed without error

Bit 3 "Job terminated with error"

- set by the system program when a job is terminated with an error
- you can read this bit and then reset it, or it is reset by the system program when bit 1 "job active" is set for a new job
- you can read this bit, to know whether a job was completed with an error

Bit 6 "Reset interface"

- you can set this bit to reset the interface
- reset by the system program as soon as the system program resets the interface

Bit 7 "Start SEND job"

- you can set this bit to start a SEND job
- reset by the system program as soon as the system program completes the SEND job

When you reset the interface, the parameters of the data link layer in the static parameter set from DW 5 onwards are read in again (refer to Section 6.3.1):

- mode
- character delay time
- end identifier

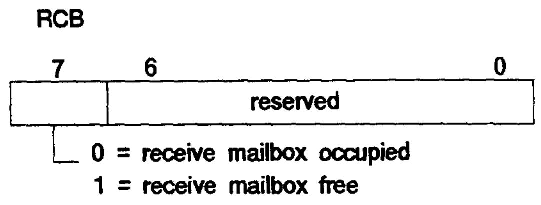
If the SCB bits 6 and 7 are sent simultaneously or if SCB bit 6 is set first followed by SCB bit 7 at an interval ≤ 10 ms (CPU 928B) or ≤ 20 ms (CPU 948), the system program first carries out the reset followed by the SEND job. If SCB bit 7 is set first followed by SCB bit 6 at an interval ≤ 10 ms (CPU 928B) or ≤ 20 ms (CPU 948), the sequence of the two jobs cannot be guaranteed. By resetting, you abort the currently active partner SEND job.

Note

You can only modify the parameters of the physical layer involving the physical transmission characteristics and the flow control (static parameter set up to and including DW 4) using a cold restart on the CPU 928B or on the CPU 948 using a cold restart or power up. Resetting the interface is not sufficient.

Receive coordination byte (RCB)

As long as the RCB bit 7 is reset, no received data are transferred to the receive mailbox.



Caution

In your user program, you should make sure that the received data are processed as quickly as possible and that the receive mailbox is enabled immediately afterwards by RCB bit 7.

Caution

During a cold restart and following power up, 00H is written to the SCB (no job active) and 80H to the RCB (receive mailbox free).

6.3.2 Getting Started

Preparations

You have already installed the interface submodule in the CPU 928B or in the CPU 948 (see Chapter 2), the CPU is plugged into the PLC rack and the power supply is off.

Procedure

We recommend the following procedure for starting data transmission with the "open driver":

Step	Action
1	Connect the CPU 982B/CPU 948 to the partner (with a standard connecting cable).
2	Switch on the power supply.
3	Transfer your user program to the CPU 928B. The user program must contain the following **data blocks** or **extended data blocks** to be able to set the SI 2 interface for data transmission with the "open driver". DX 2 DB/DX with the static parameter set DB/DX with the transmit mailbox [1] DB/DX with the receive mailbox [2] DB/DX with the SCB and RCB (only when the SCB/RCB are located in a DB/DX [3]
4	Perform a **cold restart**.
5	Prepare the partner for data transmission with the "open driver".

1) Only when the CPU 928B/CPU 948 triggers active jobs itself
2) Only when the CPU 928B/CPU 948 receives passive jobs
3) If the SCB and RCB are flags or S flag words, they already exist in the CPU 928B and do not need to be transferred.

CPU 928B

During a cold restart or following power up the system program checks that the following exist:

- DX 2
- the static parameter set
- SCB and RCB

and the correct parameter assignment in DX 2

DX 2 exists and has correct parameters

- The system program initializes the SI 2 interface for data transmission with the "open driver".
- The red LED SI 2 goes off as soon as data transmission with the "open driver" is ready for operation.
- The system program resets the SCB bit 7 (i.e. no SEND job active).
- The system program resets the SCB bit 7 (i.e. receive mailbox enabled).

↓

Data transmission with the "open driver" is ready for operation

DX 2 does not exist or does not have correct parameters

- The CPU remains in the STOP mode.
- The red LED SI 2 is lit in addition to the STOP LED.
- The system program outputs an error message to RS 3 and RS 4.

CPU 948

During a cold restart or following power up the system program checks that the following exist:

- DX 2
- the static parameter set
- SCB and RCB

and the correct parameter assignment in DX 2

DX 2 exists and has correct parameters

- The system program initializes the SI 2 interface for data transmission with the "open driver".
- The red LED SI 2 goes off as soon as data transmission with the "open driver" is ready for operation.
- The system program resets the SCB bit 7 (i.e. no SEND job active).
- The system program resets the SCB bit 7 (i.e. receive mailbox enabled).

↓

Data transmission with the "open driver" is ready for operation

DX 2 does not exist or does not have correct parameters

- The system program does not initialize the SI 2 interface for data transmission with the "open driver"
- The red LED SI 2 does not go off

6.3.3 Operation

Preparations

The SI 2 interface has parameters assigned for data transmission with the "open driver". The SI 2 LED is off. Data transmission with the "open driver" is ready for operation.

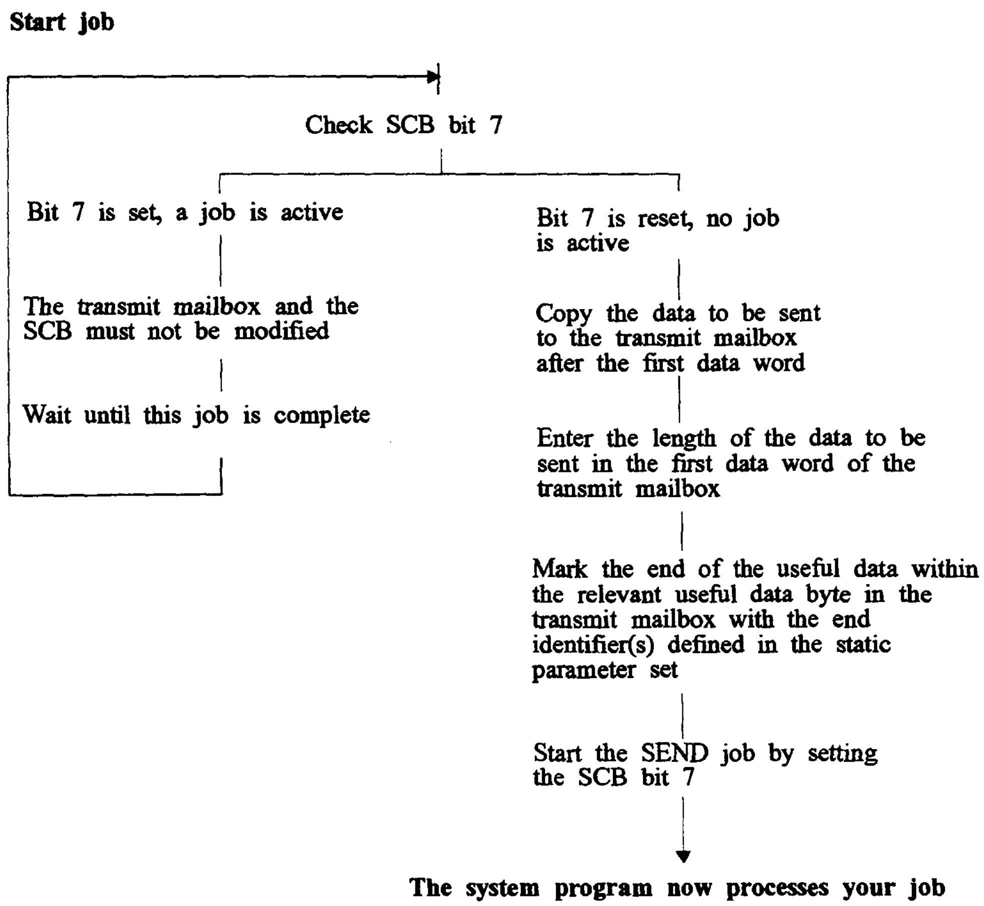
Method

The system program now checks constantly whether the send coordination byte (SCB) and receive coordination byte (RCB) exist. If they do not exist (e.g. because the DB/DX has been deleted), the system program calls OB 35. The error number is entered in ACCU 1 (see Chapter 11).

The system program now processes the jobs you have programmed in your user program, or the jobs triggered by the communications partner. The following sections explain how to program the jobs in your user program.

SEND job

If you want to program a SEND job, remember the following sequence for the various modes:



The following diagram illustrates the reactions that can occur if the SCB bit 7 is set and the system program is processing your job.

SCB bit 7 is set

- The transmit mailbox exists
  - The system program transfers the data including the end identifier
    - The job was completed without errors
      - The system program
        - transmitted the data from the transmit mailbox to the partner
        - reset the SCB bit 7
    - The job was completed with an error
      - The system program
        - called OB 35: ACCU 1 contains the error number
        - reset the SCB bit 7
- The transmit mailbox does not exist
  - The system program
    - did not process the job
    - called OB 35 and entered an error number in ACCU 1
    - reset the SCB bit 7

Note

You can only start a new SEND job when the previous job is completed.

If a SEND job was completed with an error, you must wait for approximately twice the character delay time of the communications partner before you can start a new SEND job, otherwise further errors will result at the communications partner.

Resetting the interface

You have set the SCB bit 6 and have therefore started a reset job. The following diagram shows the possible reactions.

SCB bit 6 is set

The CPU 928B/CPU 948 is currently processing a job from the partner

- the partner's job is aborted
- OB 35 has been called: ACCU 1 contains the error number of the aborted job on the partner
- the transmit and receive buffers are cleared (see note)
- the static parameters from DW 5 onwards are read and evaluated
- when the reset job is complete, the SCB bit 6 is reset

The CPU 928B/CPU 948 is not processing a job from the partner

- the static parameters from DW 5 onwards are read in and evaluated
- when the reset job is complete, the SCB bit 6 is reset

SEND job on the partner

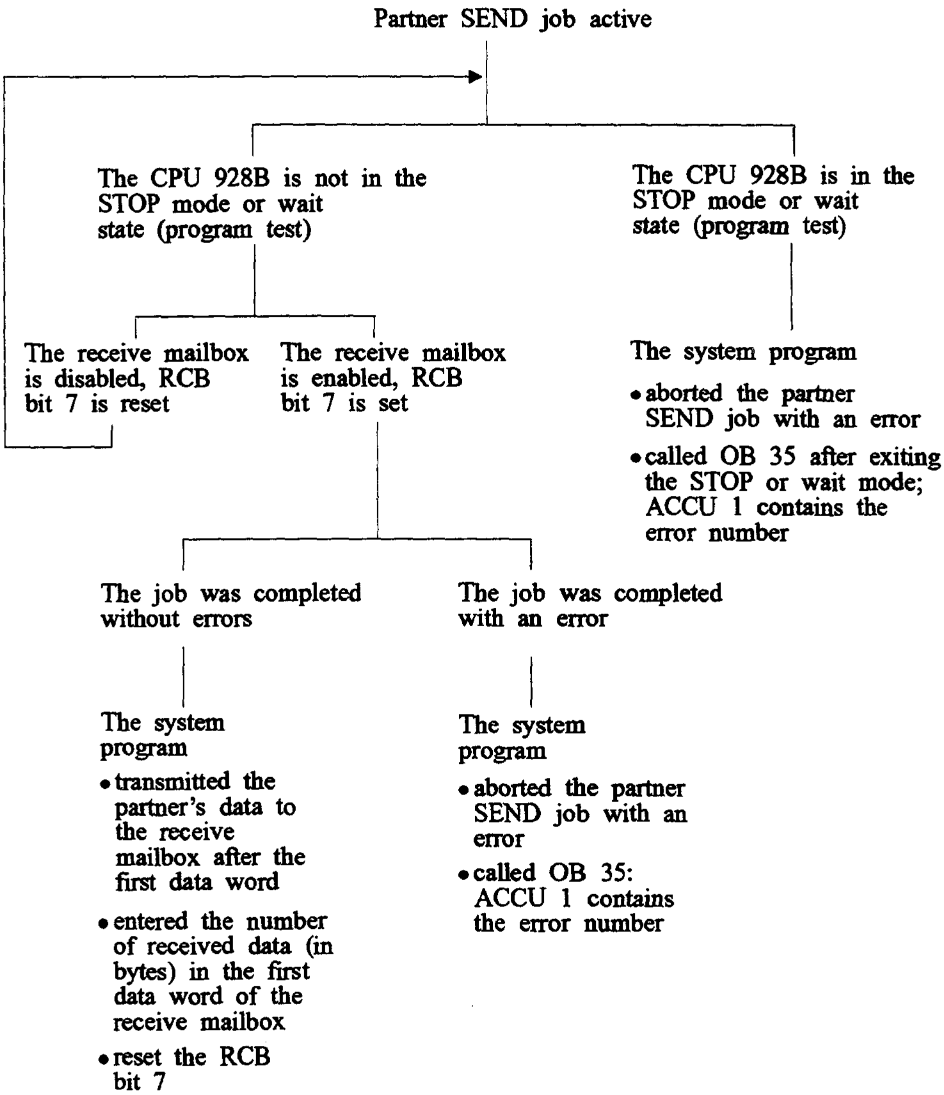
SEND jobs on the partner are not processed by the user program, but automatically by the system program in the background. The system program resets the RCB bit 7 (receive mailbox disabled) when the data have been entered completely in the receive mailbox. You can now evaluate the received data. The receive mailbox can only be overwritten when you have enabled it by setting RCB bit 7.

Caution

In your user program, make sure that the received data are processed as quickly as possible and then enable the receive mailbox immediately by setting the RCB bit 7.

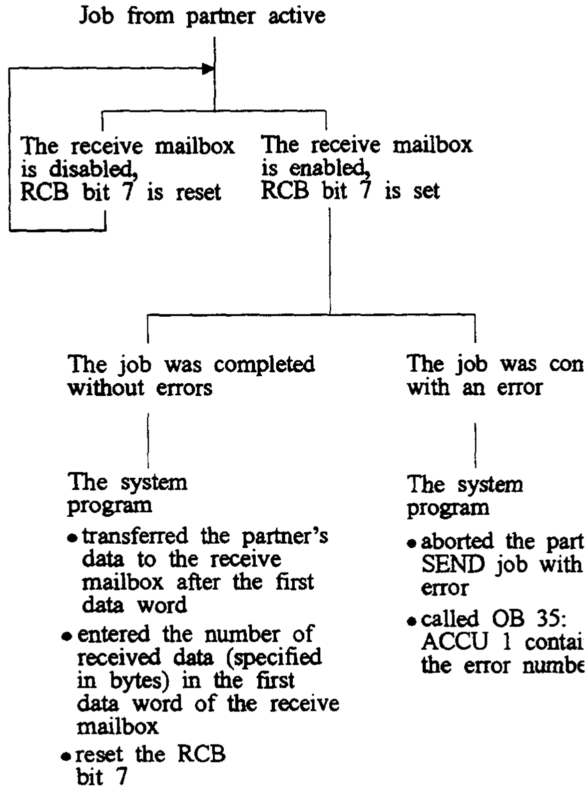
CPU 928B

The following diagram illustrates the situations that can occur while a partner SEND job is being executed in mode 1.



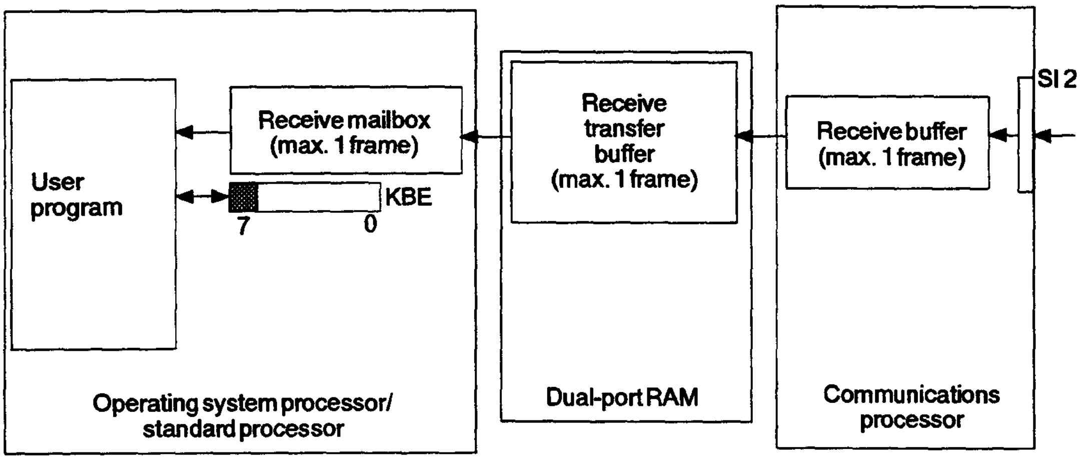
CPU 948

The following diagram illustrates the situations that can occur while a partner SEND job is being executed in mode 1.



Points to note when receiving

Only one message can ever be entered in the receive mailbox (max. 4096 bytes long), although the receive transfer buffer of the dual-port RAM and the receive buffer can both hold one further message. Once you empty the mailbox and set bit 7 (receive mailbox free) in the receive coordination byte (RCB), the next message, if it exists, is transferred from the receive transfer buffer of the dual-port RAM into the mailbox and the next message shifted from the receive buffer to the receive transfer buffer of the dual-port RAM.



Response to special operating modes

Stop mode

CPU 928B

In the "STOP" mode, user programs are not processed, i.e. you cannot start a new job. When the CPU changes from "RUN" to "STOP", the following applies to an already active job started by the user program:

- The job is stopped with an error, if there is still data to be transferred between the operating system processor and the dual-port RAM.
- The job is completed normally if there is no more data to be transferred between the operating system processor and the dual-port RAM.

If the partner triggers a job in the "STOP" mode, this is acknowledged negatively. An appropriate error message is entered in the error area of the interface (refer to Chapter 11 "Error Messages"). When the CPU changes from the "RUN" mode to the "STOP" mode, an already active job started by the partner is stopped with an error, i.e. negatively acknowledged. An appropriate error message is entered in the error area of the interface (refer to Chapter 11 "Error Messages")

CPU 948

In contrast to the CPU 928B, the CPU 948 also supports communication in "soft" stop and in the wait mode (program test).

Wait mode

CPU 928B

The CPU 928B goes into the wait mode when the PG online function "PROGRAM TEST" is active. In the wait mode user programs are not processed, i.e. you cannot start a new job. An already active job started by the user is continued and completed after the wait mode is exited.

If the partner triggers a job in the wait mode, this is acknowledged negatively. An appropriate error message is entered in the error area of the interface (refer to Chapter 11 "Error Messages"). When the CPU changes to the wait mode, an already active job started by the partner is negatively acknowledged. An appropriate error message is entered in the error area of the interface (refer to Chapter 11 "Error Messages")

CPU 948

In contrast to the CPU 928B, the CPU 948 also supports communication in "soft" stop and in the wait mode (program test).

6.3.4 Protocol Definition of the "Open Driver" in Mode 2

The following section is intended to help you if you want to connect a non-SIMATIC device using the "open driver" in mode 2 (variable user data length). It is also intended, however, as a source of reference for engineers who have to develop a protocol compatible with the "open driver" on a non-SIMATIC computer.

The following aspects of transmitting and receiving are described:

- structure of the transmitted/received data frame
- any restrictions
- basic sequence of events

The sequence of events refer to the communications processor. The heavy lines indicate the standard sequence, whereas the thinner lines represent error handling.

Transmitting

Setting up the transmit mailbox is described in Section 6.3.1 "Transmit mailbox". The sequence of the SEND job is described in Section 6.3.3 "SEND job".

In mode 2, the user data stored behind the first data word in the transmit mailbox are transmitted inclusive of the end identifier parameterized in the static parameter set.

The data frame to be transmitted is structured as shown below:

Length of the user data: N bytes

1 … N	N+1	N+2
User data - length N bytes	E1	E2

E1: first end identifier
E2: second end identifier, only exists if assigned in static parameter set

The end identifier selected in the static parameter set must be located in the transmit mailbox after the user data to be sent.

Caution
If you select flow control with XON/XOFF, the user data must not contain XON (DC1 = 11H) or XOFF (DC3 = 13H).

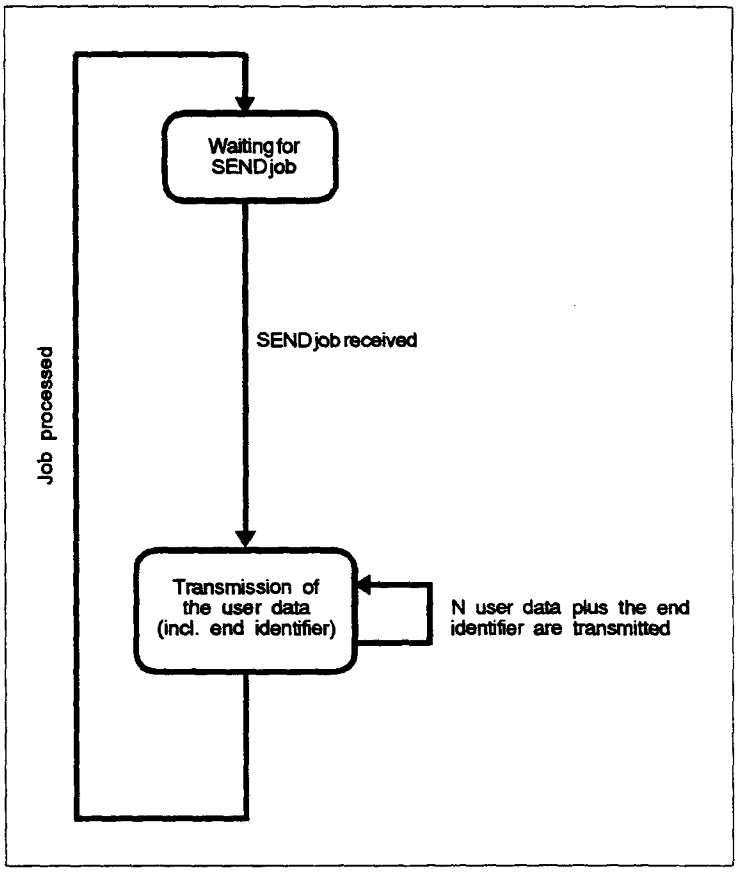


Fig. 6-8 Sequence of transmission in mode 2

Receiving

The setting up of the receive mailbox is described in Section 6.3.1 "Receive mailbox". The sequence of the partner SEND job is described in Section 6.3.3 "SEND job on the partner".

The received characters including the end identifier are entered in the receive mailbox after the user data. The data frame to be received has the following structure:

Length of the user data: N bytes

1 ... N	N+1	N+2
User data - length N bytes	E1	E2

E1: first end identifier
E2: second end identifier, (if defined in the static parameter set)

The reception of the end identifier indicates a valid message. If the character delay time (ZVZ) elapses, this indicates an invalid frame.

The end identifier must be the same as defined in the static parameter set.

Caution

Make sure that both communications partners operate with the same end identifier.

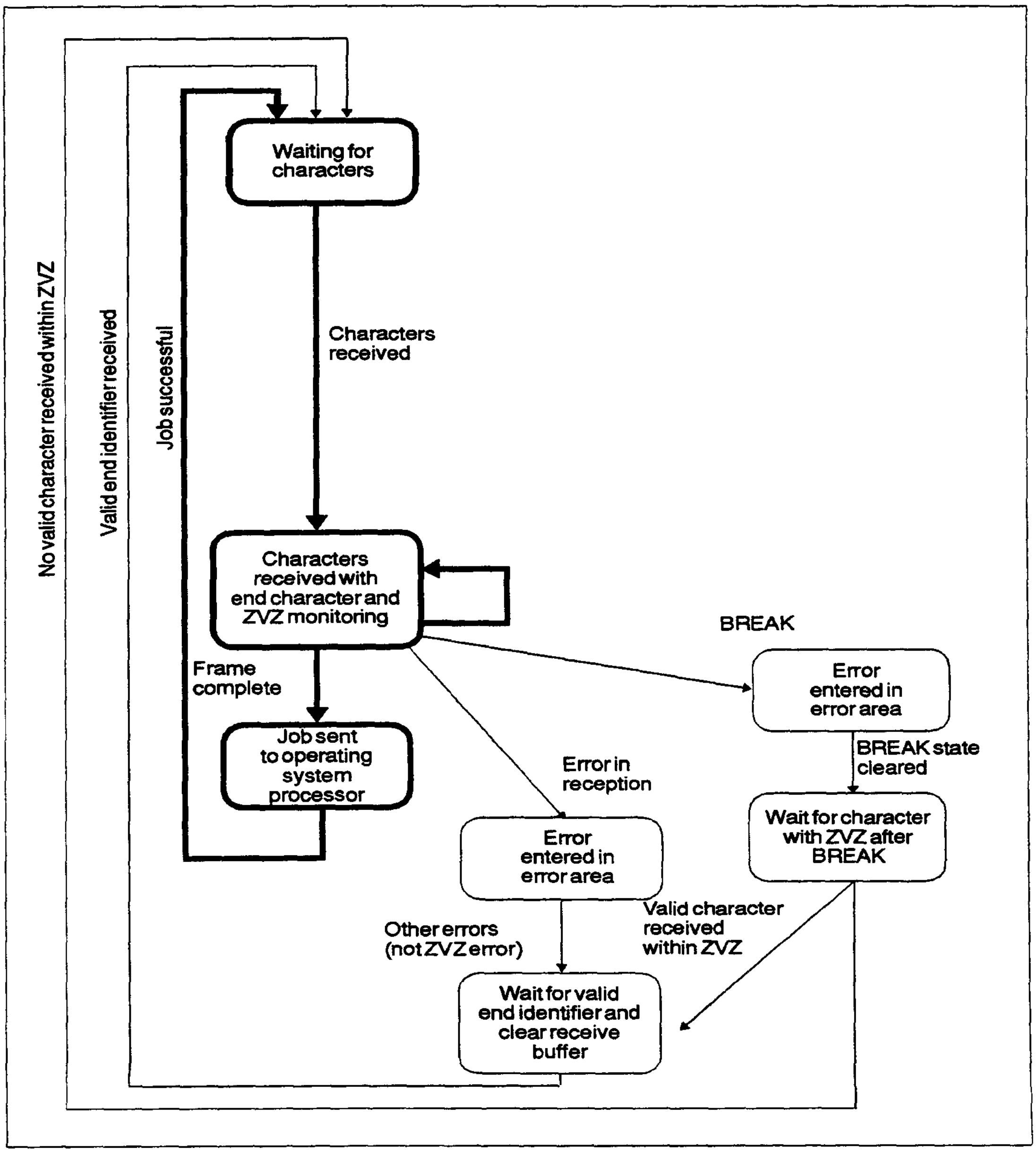


Fig. 6-9 Sequence of receiving in mode 2

6.3.5 Processing Times

This section lists the typical processing times for a SEND job.

Processing times for the CPU 928B

The processing times are subject to the following conditions:

- two CPU 928Bs are connected
- both CPUs have parameters assigned for for data transfer with the "open driver" in mode 2 (variable user data length). In the static parameter set two end identifiers and a character delay time of 220 ms are selected (even parity, 8 bits per character, 1 stop bit).

The processing time is the time
- from starting the job on the active CPU 928B (the user program sets SCB bit 7)
- until the job is completed on the active CPU 928B (the system program resets SCB bit 7)

SEND job

The times were measured
- with different transmission speeds and
- with various amounts of user data (the end identifier is not included in the user data)

No. of bytes	Transmission speed								
	110	150	300	600	1200	2400	4800	9600	19200
1	0.298s	0.218s	0.109s	0.054s	0.029s	0.019s	0.009s	0.009s	0.009s
2	0.399s	0.293s	0.146s	0.073s	0.039s	0.019s	0.019s	0.009s	0.009s
4	0.600s	0.439s	0.219s	0.109s	0.054s	0.029s	0.019s	0.010s	0.009s
8	1.000s	0.733s	0.366s	0.183s	0.092s	0.049s	0.029s	0.019s	0.010s
16	1.800s	1.319s	0.659s	0.329s	0.169s	0.089s	0.049s	0.029s	0.019s
32	3.400s	2.493s	1.246s	0.623s	0.314s	0.159s	0.089s	0.049s	0.029s
64	6.600s	4.839s	2.419s	1.209s	0.609s	0.309s	0.159s	0.089s	0.049s
128	13.000s	9.532s	4.766s	2.385s	1.199s	0.609s	0.319s	0.169s	0.090s
256	25.800s	18.920s	9.460s	4.729s	2.389s	1.209s	0.610s	0.329s	0.177s
384	38.600s	28.304s	14.160s	7.099s	3.569s	1.809s	0.929s	0.482s	0.260s
512	51.400s	37.690s	18.860s	9.442s	4.759s	2.409s	1.229s	0.639s	0.349s
1024	102.608s	75.268s	37.688s	18.880s	9.509s	4.809s	2.459s	1.289s	0.699s
2048	205.120s	150.464s	75.332s	37.760s	18.998s	9.609s	4.909s	2.559s	1.389s
3072	307.648s	225.664s	112.992s	56.646s	28.492s	14.408s	7.369s	3.849s	2.089s
4094	409.920s	300.688s	150.560s	75.492s	37.968s	19.189s	9.810s	5.119s	2.779s

Processing times for the CPU 948

The processing times are subject to the following conditions:

- two CPU 948s are connected
- both CPUs have parameters assigned for for data transfer with the "open driver" in mode 2 (variable user data length). In the static parameter set two end identifiers and a character delay time of 220 ms are selected (even parity, 8 bits per character, 1 stop bit).

The processing time is the time

- from starting the job on the active CPU 948 (the user program sets SCB bit 7)
- until the job is completed on the active CPU 948 (the system program resets SCB bit 7)

SEND job

The times were measured

- with different transmission speeds and
- with various amounts of user data (the end identifier is not included in the user data)

No. of bytes	Transmission speed								
	110	150	300	600	1200	2400	4800	9600	19200
1	0.446s	0.329s	0.179s	0.099s	0.069s	0.049s	0.049s	0.039s	0.039s
2	0.545s	0.399s	0.219s	0.119s	0.079s	0.049s	0.049s	0.030s	0.039s
4	0.745s	0.549s	0.289s	0.159s	0.097s	0.055s	0.049s	0.040s	0.039s
8	1.143s	0.839s	0.438s	0.229s	0.129s	0.075s	0.052s	0.042s	0.039s
16	1.942s	1.429s	0.729s	0.379s	0.203s	0.112s	0.072s	0.056s	0.049s
32	3.545s	2.606s	1.314s	0.669s	0.349s	0.189s	0.112s	0.079s	0.059s
64	6.745s	4.954s	2.496s	1.260s	0.649s	0.339s	0.189s	0.119s	0.079s
128	13.149s	9.648s	4.839s	2.438s	1.229s	0.630s	0.338s	0.189s	0.109s
256	25.954s	18.039s	9.537s	4.787s	2.410s	1.220s	0.639s	0.339s	0.189s
384	38.758s	28.431s	14.235s	7.137s	3.589s	1.809s	0.929s	0.489s	0.270s
512	51.570s	37.826s	18.937s	9.493s	4.769s	2.411s	1.231s	0.644s	0.359s
1024	102.797s	75.395s	37.732s	18.901s	9.485s	4.779s	2.429s	1.254s	0.668s
2048	205.254s	150.532s	75.322s	37.716s	18.914s	9.512s	4.817s	2.471s	1.295s
3072	307.707s	225.670s	112.912s	56.533s	28.343s	14.249s	7.206s	3.685s	1.925s
4094	409.958s	300.655s	150.423s	75.307s	37.748s	19.970s	9.586s	4.893s	2.548s

6.4 Mode 3 - Fixed User Data Length, Unsymmetrical

6.4.1 Parameter Assignment

The following sections explain how you can assign parameters for data transmission with the "open driver" in mode 3 and how you should structure the required data blocks.

You have two options:

- either assign parameters directly in STEP 5 by inputting them into data blocks
- or enter the parameters via the menus in the COM PP Parameter Assignment Software and then transfer the data blocks you have created to the CPU using the STEP 5 basic package. Working with COM PP is described in Chapter 3.

Assigning parameters to DX 2

DX 2 contains the link type. It also contains the pointers to the required parameters and parameter sets. The following table shows the values you can use when assigning parameters to DX 2. The assignment in DX 2 begins at DW 0 (absolute). All the numerical values are in hex.

DW	Parameter	Significance
0	4D41	MA
1	534B	SK
2	5832	X2
Link type		
3	0030	Data transmission with the "open driver"
Pointer to static parameter set		
4	44xx or 58xx	DB no. xx or[4] DX no. xx
5	xxxx	from DW no. xxxx
-		
6	0000	reserved
7	0000	reserved
Pointer to transmit mailbox [1]		
8	44xx or 58xx	DB no. xx or [4] DX no. xx
9	xxxx	from DW no. xxxx
10	xxxx	Length of transmit mailbox in words (range of values 0002H to 0801H)
Pointer to receive mailbox [2]		
11	44xx or 58xx	DB no. xx or [4] DX no. xx
12	xxxx	from DW no. xxxx
13	xxxx	Length of receive mailbox in words (range of values 0002H to 0801H)
Pointer to SCB/RCB [3]		
14	4D00 or 5300 or 44xx or 58xx	Flag or [4] S flag or DB no. xx or DX no. xx
15	xxxx	FW no., SW no. or DW no. xxxx

1) These parameters are not necessary if the CPU 928B/CPU 948 is purely passive. In this case, enter the value 0000 in data words DW 8 to DW 10.

2) These parameters are not necessary if the CPU 928B/CPU 948 is purely active, i.e. does not receive data. If this is the case, enter the value 0000 in data words DW 11 to DW 13.

3) The locations of SCB and RCB must be selected together. See also Section 6.4.1.

4) Range of values for xx from 3 to FF (3 - 255)

Example

In this example you can see how to assign the following basic parameters to a DX 2:

- The static parameter set is in data block 10 from data word 4 onwards.
- The transmit mailbox is in data block 6 from data word 2 onwards, length 10 words.
- The receive mailbox is in data block 8 from data word 1 onwards, length 6 words.
- The send and receive coordination bytes (SCB and RCB) are in flag word 6.

With these parameters, DX 2 appears as below:

DX 2

0:	KH = 4D41;	
1:	KH = 534B;	'MASKX2'
2:	KH = 5832;	
3:	KH = 0030;	**LINK TYPE** "open driver"
4:	KH = 440A;	**STATIC PARAMETER SET** IN DB 10
5:	KH = 0004;	FROM DATA WORD 4
6:	KH = 0000;	reserved
7:	KH = 0000;	
8:	KH = 4406;	**TRANSMIT MAILBOX** IN DATA BLOCK 6
9:	KH = 0002;	FROM DATA WORD 2
10:	KH = 000A;	LENGTH 10 WORDS
11:	KH = 4408;	**RECEIVE MAILBOX** IN DATA BLOCK 8
12:	KH = 0001;	FROM DATA WORD 1
13:	KH = 0006;	LENGTH 6 WORDS
14:	KH = 4D00;	**COORDINATION BYTES** IN FLAG AREA
15:	KH = 0006;	SCB FLAG BYTE 6, RCB FLAG BYTE 7
16:	KH = xxxx;	no longer belongs to the parameter assignment for DX 2, content irrelevant, does not need to exist
:		
:		

Static parameter set

In the static parameter set, you must enter the parameters for the physical layer as well as specific transmission parameters.

The following table shows the values that you can enter in a data block or extended data block for the static parameter set. The parameter assignment begins at the DW specified in DX 2.

All numerical values are shown in hexadecimal format.

DW (rel.)	Range of values	Significance
Transmission speed		
0	0001	110 bps
	0002	150 bps
	0003	300 bps
	0004	600 bps
	0005	1200 bps
	0006	2400 bps
	0007	4800 bps
	0008	9600 bps
	0009	19200 bps (only with V.24 and RS422-A/485 submodules)
Parity		
1	0000	no
	0001	odd
	0002	even
Bits per character		
2	0006	6 bits per character
	0007	7 bits per character
	0008	8 bits per character
Stop bits		
3	0001	1 stop bit
	0002	1 + 1/2 stop bits
	0003	2 stop bits
Flow control		
4	0000	none
	0001	XON/XOFF
	0002	RTS/CTS
Mode		
5	0003	fixed user data length, unsymmetrical
-		
6	0000	reserved
Character delay time		
7	0001 to FFFF	monitoring time 0.01 sec. to 655.35 sec. (unit: 10 ms)

DW (rel.)	Range of values	Significance
8	0000	reserved
9	0000	reserved
10	0000	reserved
-		
11	0000	reserved
12	0000	reserved
Fixed frame length for reception		
13	0001 to 1000	Frame length in bytes

6

Note

To avoid corrupting frames, it is advisable to operate with flow control.

Meaning of the parameters

Transmission speed

Specified in bps.

Parity

Number of bits with the value "1" in a string of information bits with a uniform length (e.g. 1 byte). Depending on the value, the parity is "even" or "odd". To check the parity, the string of information bits is extended by a further bit, the parity bit ("0" or "1") which when added to the information bits produces a selectable parity state.

Bits per character

Number of data bits used to form a character.

Stop bits

Duration of the stop bits, relative to the time required to transmit an information bit. The stop bits follow each character transmitted in a start-stop transmission.

Flow control

A mechanism with which the transmission (CPU or partner) and reception (partner or CPU) of data is synchronized when the data source operates faster than the data sink.
The control of the data flow using flow control is transparent for the user (with the exception of the required parameter assignment). This means that if you use flow control 0001 (XON/XOFF, control characters DC1 and DC3), the user data must not contain these control characters.
If you use flow control 0002 (RTS/CTS) you must also wire signals RTS and CTS (see also Section 5.8 of the S5-135U/155U System Manual).

Mode

Version of the possible methods of data transmission

Mode 3 (fixed user data length, unsymmetrical)

Sending

When sending n bytes of user data, you must specify the number n of user data to be transmitted in the first word of the transmit mailbox. This specified length does not include the word containing the length specification itself.

If you only want to transmit, you must nevertheless assign the relative DW 13 in the static parameter set (0000H < DW 13 ≤ 1000H).

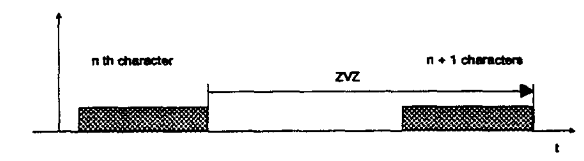
Receiving

When receiving, the user data up to the assigned fixed frame length are accepted, (i.e. the relative DW 13 must be assigned in the static parameter set). The system program enters the number n of received useful bytes in the first word of the receive mailbox. (Must correspond to the "fixed frame length for reception"). This length does not include the word containing the length specification.

If the character delay time elapses during the reception of data, reception is terminated. The frame is discarded.

Character delay time (ZVZ)

The maximum permitted time interval between two received characters (see following table).



Transmission speed	Min. permitted ZVZ
110 bps	120 ms
150 bps	90 ms
300 bps	50 ms
600 bps	30 ms
1200 bps	20 ms
2400 bps	10 ms
4800 bps	10 ms
9600 bps	10 ms
19200 bps	10 ms

Fixed frame length for reception

When characters are received, the frame end is detected based on the selected fixed frame length.

Example

The example indicates how to write a static parameter set in DB 10 from DW 4 for a point-to-point link in transmission mode 3.

DB 10

0:	KH = xxxx;	does not belong to the static parameter set,
1:	KH = xxxx;	content irrelevant
2:	KH = xxxx;	
3:	KH = xxxx;	
4:	KH = 0007;	**transmission speed** = 4800 bps
5:	KH = 0001;	**parity** = 1 = odd
6:	KH = 0006;	**bits per character** = 6
7:	KH = 0002;	**stop bits** = 1 + 1/2
8:	KH = 0000;	**no flow control**
9:	KH = 0002;	**mode** = 3 = fixed user data length, unsymmetrical
10:	KH = 0000;	reserved
11:	KH = 0037;	**character delay time** = 55 x 0.01sec (=550 ms)
12:	KH = 0000;	
13:	KH = 0000;	reserved
14:	KH = 0000;	
15:	KH = 0000;	
16:	KH = 0000;	
17:	KH = 0009;	fixed frame length for reception: 9 bytes
18:	KH = yyyy;	no longer required in mode 3, content irrelevant,
:		does not need to exist
:		

Note

For data transmission with the "open driver" the parameters of the physical layer on the communications partner must match those of the CPU 928B or of the CPU 948.

Transmit mailbox

The transmit mailbox is located in a data block (DB) or extended data block (DX).

The transmit mailbox contains the data to be sent as well as the length specification in the first word (in bytes). This means that you can decide how many bytes of data are to be sent to the communications partner for each job. The specified length does not include the word containing the length specification itself. The length specification is not transmitted.

The length of the data to be transmitted must not exceed the length of the transmit mailbox minus one word (see above). The length of the transmit mailbox is a maximum of 2049 words (4098 bytes). You specify this in DX 2.

Remember to specify the length of the transmit mailbox in **words** in DX2, but the length of the data to be transmitted in **bytes** in the first word of the transmit mailbox. Select the length of the transmit mailbox as suitable for your particular application.

Example: transmit mailbox

The example illustrates how to assign parameters for a transmit mailbox in data block 6 from data word 2 onwards with a length of 10 words. You want to transmit 9 bytes.

DB 6

0:	KH = xxxx;	does not belong to transmit mailbox, content irrelevant
1:	KH = xxxx;	
2:	KH = 0009;	start of transmit mailbox, length in bytes
3:	KH = 0102;	data to be transmitted: 0102
4:	KH = 0304;	data to be transmitted: 0304
5:	KH = 0506;	data to be transmitted: 0506
6:	KH = 0708;	data to be transmitted: 0708
7:	KH = 09yy;	data to be transmitted: 09, yy is not transmitted
8:	KH = yyyy;	the content of the transmit mailbox is no longer sent
9:	KH = yyyy;	from this point
10:	KH = yyyy;	
11:	KH = yyyy;	
12:	KH = xxxx;	does not belong to the transmit mailbox, content
:		irrelevant, does not need to exist
:		

Receive mailbox

The receive mailbox is in a data block (DB) or extended data block (DX).

The receive mailbox contains the received data after the first data word. The system program enters the number n (in bytes) of received user data bytes in the first word of the receive mailbox (must correspond to the parameter "fixed message length for reception" in the static parameter set). This specified length does not include the word containing the length specification itself. The number of received data can be determined by the parameter "fixed message length for reception" in the static parameter set. Make sure that the communications partner really transmits the number of bytes selected in the static parameter set, otherwise the received data are discarded.

The "fixed message length for reception" must not exceed the length of the receive mailbox minus one word (see above). The length of the receive mailbox is a maximum of 2049 words (4098 bytes). You specify the length in DX 2.

Remember to specify the length of the receive mailbox in **words**, but the parameter "fixed frame length for reception" in **bytes.**

Example: receive mailbox

The example illustrates the structure of a receive mailbox in data block 8 from data word 1 onwards with a length of 6 words. In the static parameter set, the "fixed message length for reception" was selected as 9 bytes.

DB 8

0:	KH = xxxx;	does not belong to receive mailbox, content irrelevant
1:	KH = 0009;	start of the receive mailbox, length specified in bytes
2:	KH = 0102;	received data: 0102
3:	KH = 0304;	received data: 0304
4:	KH = 0506;	received data: 0506
5:	KH = 0708;	received data: 0708
6:	KH = 09yy;	received data: 09, yy was not received
7:	KH = xxxx;	does not belong to the receive mailbox, content irrelevant, does not need to exist
:		
:		

Send and receive coordination bytes (SCB and RCB)

With the SCB you can start the following jobs:

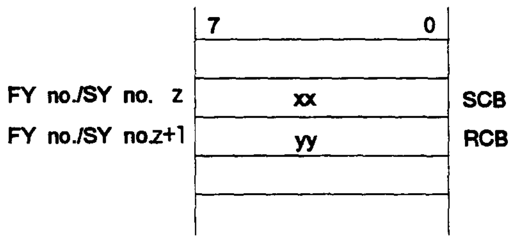
- SEND job
- reset the "open driver"

With the RCB you can perform the following

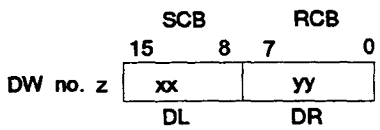
- enable and disable the receive mailbox

The coordination bytes can be flags or S flags or be located in a data block or extended data block. Flags and S flags already exist on the CPU 928B or CPU 948, you must set up data blocks and extended data blocks yourself.

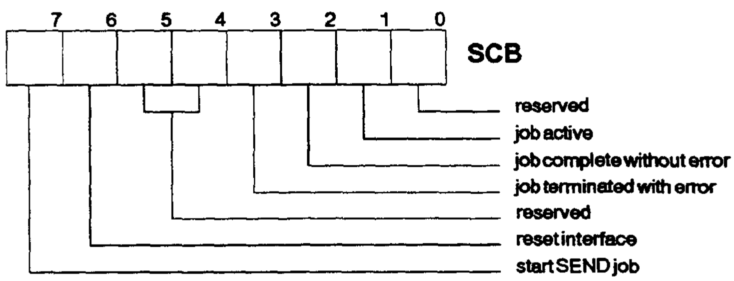
Representation in the flag or S flag area



Representation in a data block or extended data block



Send coordination byte (SCB) The send coordination byte is structured as follows:



The individual bits or bit groups, providing they are not reserved, are described below.

Bit 1 "Job active"

- set by the system program when a job starts
- reset by the system program when a job is completed
- you can read this bit, to know whether there is any point in starting a new job

Bit 2 "Job complete without error"

- set by the system program when a job is completed without error
- you can read this bit and then reset it, or it is reset by the system program when bit 1 "job active" is set for a new job
- you can read this bit, to know whether a job was completed without error

Bit 3 "Job terminated with error"

- set by the system program when a job is terminated with an error
- you can read this bit and then reset it, or it is reset by the system program when bit 1 "job active" is set for a new job
- you can read this bit, to know whether a job was completed with an error

Bit 6 "Reset interface"

- you can set this bit to reset the interface
- reset by the system program as soon as the system program resets the interface

Bit 7 "Start SEND job"

- you can set this bit to start a SEND job
- reset by the system program as soon as the system program completes the SEND job

When you reset the interface, the parameters of the data link layer in the static parameter set (from DW 5 onwards) are read in again (refer to Section 6.4.1):

- mode
- character delay time
- fixed frame length for reception

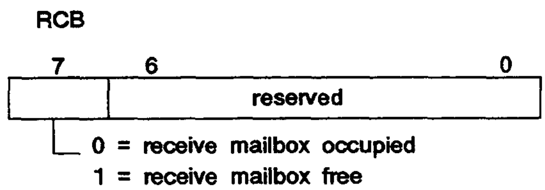
If the SCB bits 6 and 7 are sent simultaneously or if SCB bit 6 is set first followed by SCB bit 7 at an interval ≤ 10 ms (CPU 928B) or ≤ 20 ms (CPU 948), the system program first carries out the reset followed by the SEND job. If SCB bit 7 is set first followed by SCB bit 6 at an interval ≤ 10 ms (CPU 928B) or ≤ 20 ms (CPU 948), the sequence of the two jobs cannot be guaranteed. By resetting, you abort the currently active partner SEND job.

Note

You can only modify the parameters of the physical layer involving the physical transmission characteristics and the flow control (static parameter set up to and including DW 4) using a cold restart on the CPU 928B or on the CPU 948 using a cold restart or power up. Resetting the interface is not sufficient.

Receive coordination byte (RCB)

As long as the RCB bit 7 is reset, no received data are transferred to the receive mailbox.



Caution

In your user program, you should make sure that the received data are processed as quickly as possible and that the receive mailbox is enabled immediately afterwards by RCB bit 7.

Caution

During a cold restart and following power up, 00H is written to the SCB (no job active) and 80H to the RCB (receive mailbox free).

6.4.2 Getting Started

Preparations

You have already installed the interface submodule in the CPU 928B or in the CPU 948 (see Chapter 2), the CPU is plugged into the PLC rack and the power supply is off.

Procedure

We recommend the following procedure for starting data transmission with the "open driver":

Step	Action
1	Connect the CPU 982B/CPU 948 to the partner (with a standard connecting cable).
2	Switch on the power supply.
3	Transfer your user program to the CPU. The user program must contain the following **data blocks** or **extended data blocks** to be able to set the SI 2 interface for data transmission with the "open driver". DX 2 DB/DX with the static parameter set DB/DX with the transmit mailbox [1)] DB/DX with the receive mailbox [2)] DB/DX with the SCB and RCB (only when the SCB/RCB are located in a DB/DX [3)]
4	Perform a **cold restart**.
5	Prepare the partner for data transmission with the "open driver".

1) Only when the CPU 928B/CPU 948 triggers active jobs itself
2) Only when the CPU 928B/CPU 948 receives passive jobs
3) If the SCB and RCB are flags or S flag words, they already exist in the CPU 928B and do not need to be transferred.

6

CPU 928B

During a cold restart or following power up the system program checks that the following exist:

- DX 2,
- the static parameter set
- SCB and RCB

and the correct parameter assignment in DX 2

DX 2 exists and has correct parameters

- The system program initializes the SI 2 interface for data transmission with the "open driver".
- The red LED SI 2 goes off as soon as data transmission with the "open driver" is ready for operation.
- The system program resets the SCB bit 7 (i.e. no SEND job active).
- The system program sets the SCB bit 7 (i.e. receive mailbox enabled).

↓

Data transmission with the "open driver" is ready for operation

DX 2 does not exist or does not have correct parameters

- The CPU remains in the STOP mode.
- The red LED SI 2 is lit in addition to the STOP LED.
- The system program outputs an error message to RS 3 and RS 4.

CPU 948

During a cold restart or following power up the system program checks that the following exist:

- DX 2
- the static parameter set
- SCB and RCB

and the correct parameter assignment in DX 2

DX 2 exists and has correct parameters

- The system program initializes the SI 2 interface for data transmission with the "open driver".
- The red LED SI 2 goes off as soon as data transmission with the "open driver" is ready for operation.
- The system program resets the SCB bit 7 (i.e. no SEND job active).
- The system program resets the RCB bit 7 (i.e. receive mailbox enabled).

↓

Data transmission with the "open driver" is ready for operation

DX 2 does not exist or does not have correct parameters

- The system program does not initialize the SI 2 interface for data transmission with the "open driver"
- The red LED SI 2 does not go off

6.4.3 Operation

Preparations

The SI 2 interface has parameters assigned for data transmission with the "open driver". The SI 2 LED is off. Data transmission with the "open driver" is ready for operation.

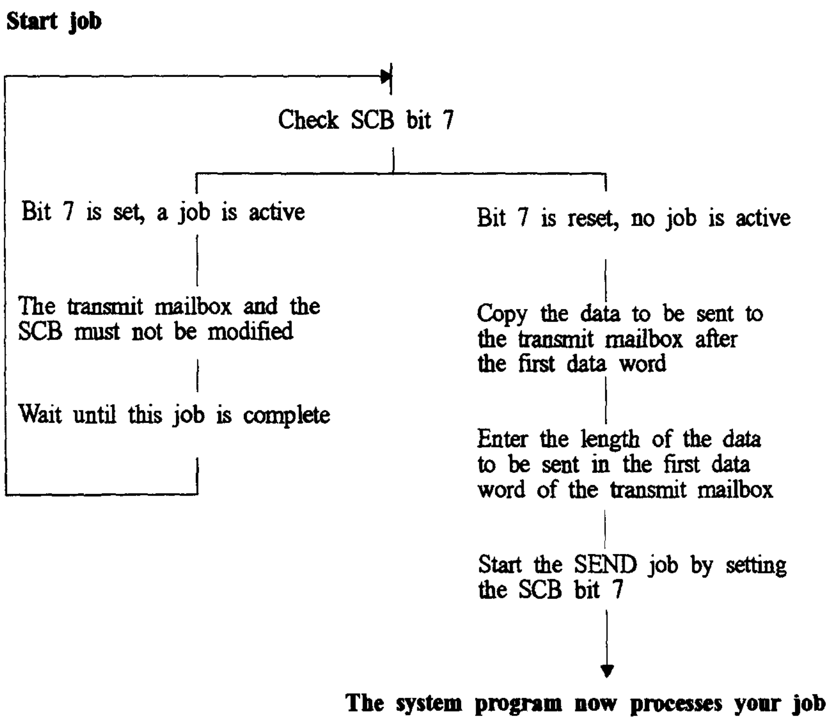
Method

The system program now checks constantly whether the send coordination byte (SCB) and receive coordination byte (RCB) exist. If they do not exist (e.g. because the DB/DX has been deleted), the system program calls OB 35. The error number is entered in ACCU 1 (see Chapter 11).

The system program now processes the jobs you have programmed in your user program, or the jobs triggered by the communications partner. The following sections explain how to program the jobs in your user program.

SEND job

If you want to program a SEND job, remember the following sequence for the various modes:



The following diagram illustrates the reactions that can occur if the SCB bit 7 is set and the system program is processing your job.

SCB bit 7 is set

The transmit mailbox exists

The system program transfers the data specified by the length word (first word in the transmit mailbox). The length word itself is not transmitted.

The error job was completed without errors

The system program

- transmitted the data from the transmit mailbox to the partner
- reset the SCB bit 7

The job was completed with an error

The system program

- called OB 35: ACCU 1 contains the error number
- reset the SCB bit 7

The transmit mailbox does not exist

The system program

- did not process the job
- called OB 35 and entered an error number in ACCU 1
- reset the SCB bit 7

Note

You can only start a new SEND job when the previous job is completed.

After a reception error the CPU 928B/CPU 948 can only synchronize itself immediately in the next frame when there is a pause in transmission by the partner of at least twice the character delay time (ZVZ).

Resetting the interface

You have set the SCB bit 6 and have therefore started a reset job. The following diagram shows the possible reactions.

SCB bit 6 is set

The CPU 928B/CPU 948 is currently processing a job from the partner

- the partner's job is aborted
- OB 35 has been called: ACCU 1 contains the error number of the aborted job on the partner
- the transmit and receive buffers are cleared (see note)
- the static parameters from DW 5 onwards are read and evaluated
- when the reset job is complete, the SCB bit 6 is reset

The CPU 928B/CPU 948 is not processing a job from the partner

- the static parameters from DW 5 onwards are read in and evaluated
- when the reset job is complete, the SCB bit 6 is reset

Note

A reset job can mean termination in the middle of the protocol. To avoid errors resulting from this, you should wait for approximately twice the character delay time of the communications partner before starting the next SEND job.

SEND job on the partner

SEND jobs on the partner are not processed by the user program, but automatically by the system program in the background. The system program resets the RCB bit 7 (receive mailbox disabled) when the data have been entered completely in the receive mailbox. You can now evaluate the received data. The receive mailbox can only be overwritten when you have enabled it by setting RCB bit 7.

Caution

In your user program, make sure that the received data are processed as quickly as possible and then enable the receive mailbox immediately by setting the RCB bit 7.

CPU 928B

The following diagram illustrates the situations that can occur while a partner SEND job is being executed in mode 1.

Partner SEND job active

The CPU 928B is not in the STOP mode or wait state (program test)

The receive mailbox is disabled, RCB bit 7 is reset

The receive mailbox is enabled, RCB bit 7 is set

The job was completed without errors

The system program
- transmitted the partner's data to the receive mailbox after the first data word
- entered the number of received data (in bytes) in the first data word of the receive mailbox
- reset the RCB bit 7

The job was completed with an error

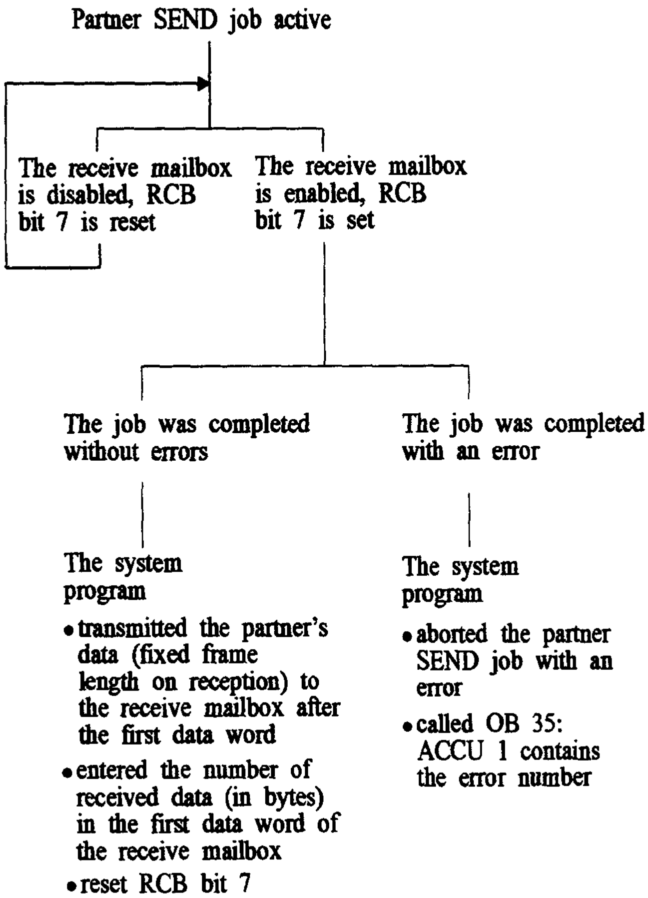
The system program
- aborted the partner SEND job with an error
- called OB 35: ACCU 1 contains the error number

The CPU 928B is in the STOP mode or wait state (program test)

The system program
- aborted the partner SEND job with an error
- called OB 35 after exiting the STOP or wait mode: ACCU 1 contains the error number

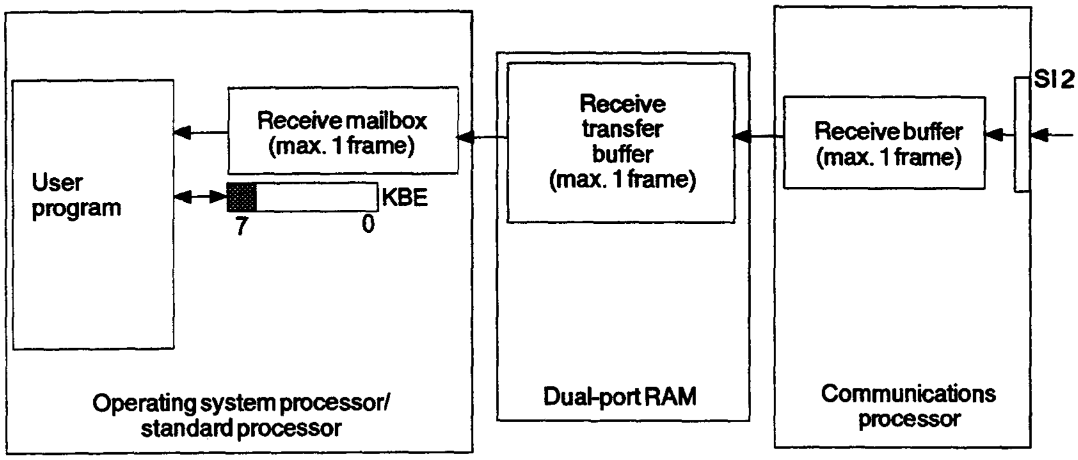
CPU 948

The following diagram illustrates the situations that can occur while a partner SEND job is being executed in mode 1.



Points to note when receiving

Only one message can ever be entered in the receive mailbox (max. 4096 bytes long), although the receive transfer buffer of the dual-port RAM and the receive buffer can both hold one further message. Once you empty the mailbox and set bit 7 (receive mailbox free) in the receive coordination byte (RCB), the next message, if it exists, is transferred from the receive transfer buffer of the dual-port RAM into the mailbox and the next message shifted from the receive buffer to the receive transfer buffer of the dual-port RAM.



Response to special operating modes

Stop mode

CPU 928B

In the "STOP" mode, user programs are not processed, i.e. you cannot start a new job. When the CPU changes from "RUN" to "STOP", the following applies to an already active job started by the user program:

- The job is stopped with an error, if there is still data to be transferred between the operating system processor and the dual-port RAM.
- The job is completed normally if there is no more data to be transferred between the operating system processor and the dual-port RAM.

If the partner triggers a job in the "STOP" mode, this is acknowledged negatively. An appropriate error message is entered in the error area of the interface (refer to Chapter 11 "Error Messages"). When the CPU changes from the "RUN" mode to the "STOP" mode, an already active job started by the partner is stopped with an error, i.e. negatively acknowledged. An appropriate error message is entered in the error area of the interface (refer to Chapter 11 "Error Messages")

CPU 948

In contrast to the CPU 928B, the CPU 948 also supports communication in "soft" stop and in the wait mode (program test).

Wait mode

CPU 928B

The CPU 928B goes into the wait mode when the PG online function "PROGRAM TEST" is active. In the wait mode user programs are not processed, i.e. you cannot start a new job. An already active job started by the user is continued and completed after the wait mode is exited.

If the partner triggers a job in the wait mode, this is acknowledged negatively. An appropriate error message is entered in the error area of the interface (refer to Chapter 11 "Error Messages"). When the CPU changes to the wait mode, an already active job started by the partner is negatively acknowledged. An appropriate error message is entered in the error area of the interface (refer to Chapter 11 "Error Messages")

CPU 948

In contrast to the CPU 928B, the CPU 948 also supports communication in "soft" stop and in the wait mode (program test).

6

6.4.4 Protocol Definition of the "Open Driver" in Mode 3

The following section is intended to help you if you want to connect a non-SIMATIC device using the "open driver" in mode 3 (fixed user data length, unsymmetrical). It is also intended, however, as a source of reference for engineers who have to develop a protocol compatible with the "open driver" on a non-SIMATIC computer.

The following aspects of transmitting and receiving are described:

- structure of the transmitted/received data frame
- any restrictions
- basic sequence of events

The sequence of events refer to the communications processor. The heavy lines indicate the standard sequence, whereas the thinner lines represent error handling.

Transmitting

Setting up the transmit mailbox is described in Section 6.4.1 "Transmit mailbox". The sequence of the SEND job is described in Section 6.4.3 "SEND job".

Only the user data are transmitted. The length information contained in the first word of the transmit mailbox is not transmitted.

The data frame to be transmitted is structured as shown below:

Length of the user data: N bytes

1 … N
User data - length N bytes

Caution

If you select flow control with XON/XOFF, the user data must not contain XON (DC1 = 11H) or XOFF (DC3 = 13H).

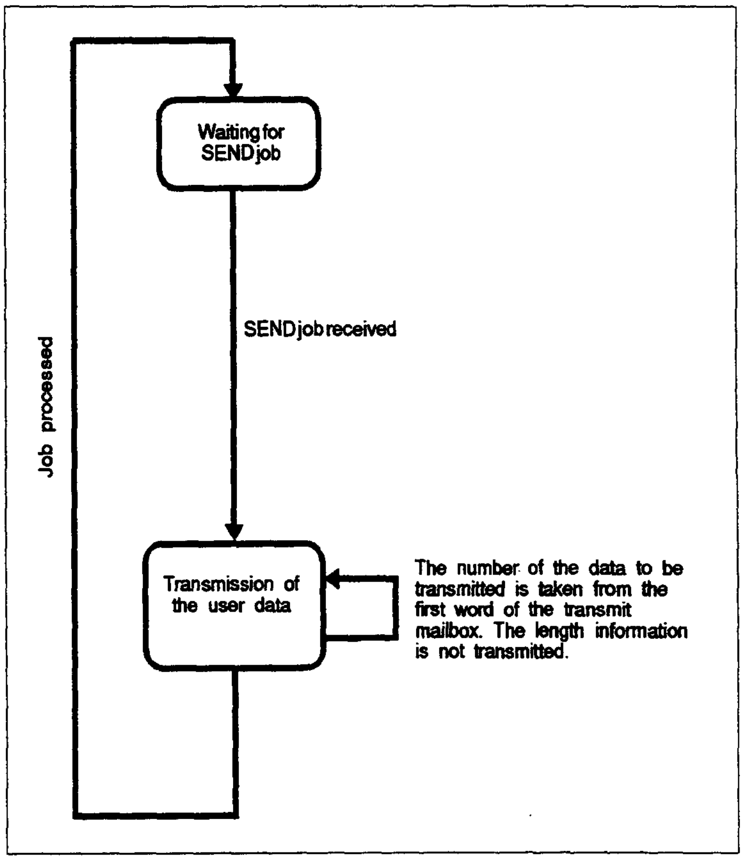


Fig. 6-10 Sequence of transmission in mode 3

Receiving

The setting up of the receive mailbox is described in Section 6.4.1 "Receive mailbox". The sequence of the partner SEND job is described in Section 6.4.3 "Send job on the partner".

The frame length in bytes is fixed by the length specified in the special static parameters. The receive mailbox contains the user data after the first data word. The first data word of the receive mailbox contains the number (in bytes) of the received user data. A frame was transmitted completely and correctly when during the reception of the number of characters defined in the special static parameters, neither the character delay time (ZVZ) elapsed (gap between characters too large) nor a transmission error of the physical layer occurred.

The data frame to be received has the following structure:

Length of the user data: N bytes

1	N
User data - length N bytes	

If the character delay time (ZVZ) elapses, the frame is invalid.

The user data length is taken from the static parameter set (DW 13).

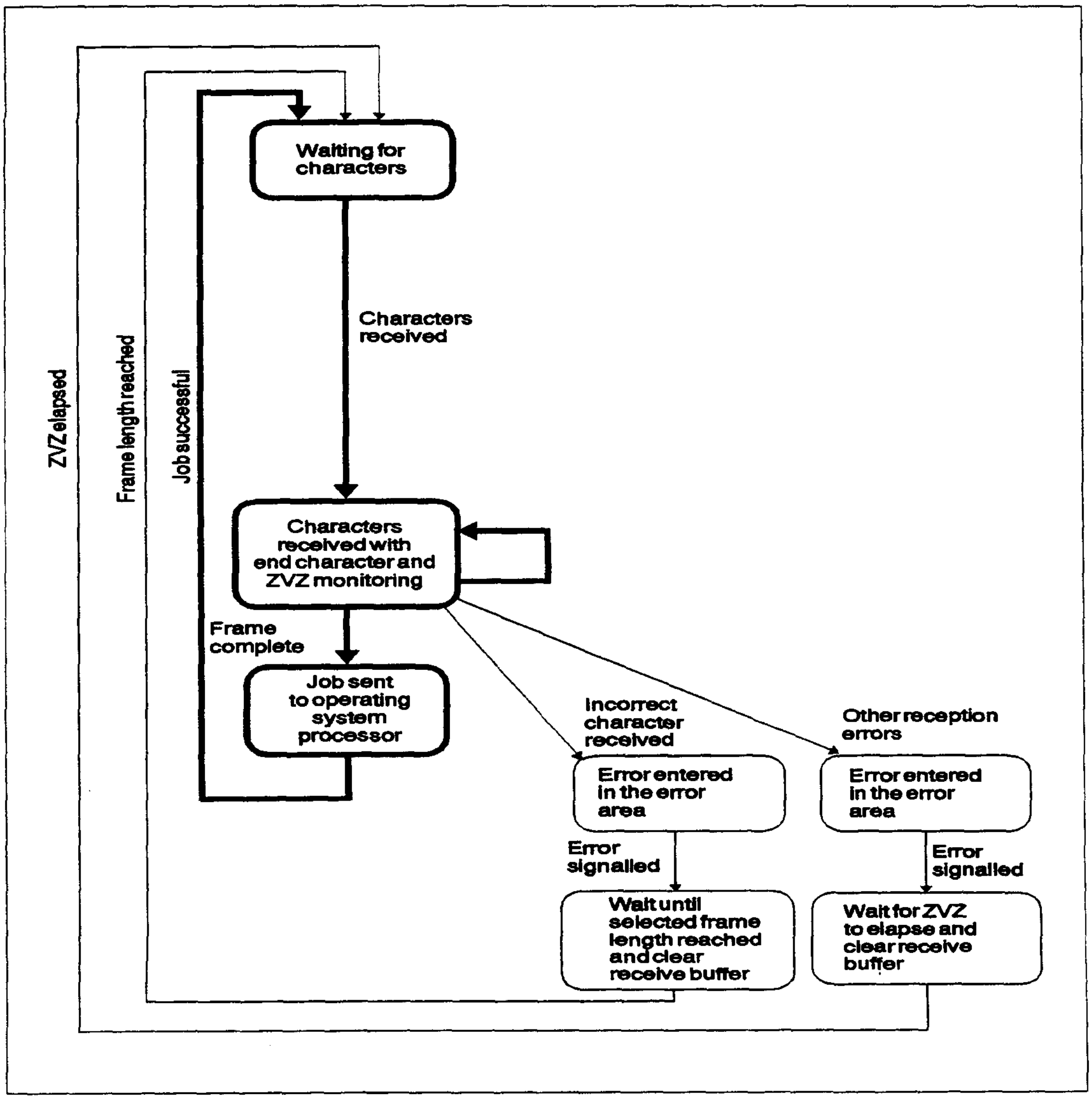


Fig. 6-11 Sequence of receiving in mode 3

6.4.5 Processing Times

This section lists the typical processing times for a SEND job.

Processing times for the CPU 928B

The processing times are subject to the following conditions:

- two CPU 928Bs are connected
- The active CPU has parameters assigned for for data transfer with the "open driver" in mode 3 (fixed user data length, unsymmetrical). In the static parameter set a character delay time of 220 ms is selected (even parity, 8 bits per character, 1 stop bit).
- The passive CPU has parameters assigned for for data transfer with the "open driver" in mode 1 (fixed user data length). In the static parameter set a character delay time of 220 ms is selected (even parity, 8 bits per character, 1 stop bit).

The processing time is the time
- from starting the job on the active CPU 928B (the user program sets SCB bit 7)
- until the job is completed on the active CPU 928B (the system program resets SCB bit 7)

6

SEND job

The times were measured
- with different transmission speeds and
- with various amounts of user data

No. of bytes	Transmission speed								
	110	150	300	600	1200	2400	4800	9600	19200
1	0.336s	0.296s	0.276s	0.266s	0.266s	0.266s	0.266s	0.266s	0.257s
2	0.339s	0.299s	0.279s	0.269s	0.269s	0.269s	0.269s	0.269s	0.259s
4	0.632s	0.509s	0.379s	0.319s	0.289s	0.279s	0.269s	0.269s	0.259s
8	1.032s	0.799s	0.529s	0.393s	0.329s	0.299s	0.279s	0.274s	0.259s
16	1.831s	1.389s	0.819s	0.539s	0.399s	0.338s	0.299s	0.279s	0.269s
32	3.432s	2.565s	1.409s	0.829s	0.549s	0.409s	0.339s	0.299s	0.269s
64	6.632s	4.919s	2.586s	1.419s	0.839s	0.558s	0.409s	0.399s	0.289s
128	13.032s	9.602s	4.926s	2.589s	1.429s	0.849s	0.559s	0.409s	0.329s
256	25.830s	18.994s	9.622s	4.939s	2.599s	1.438s	0.849s	0.559s	0.339s
384	38.634s	28.377s	14.319s	7.289s	3.779s	2.019s	1.139s	0.708s	0.479s
512	51.336s	37.696s	18.985s	9.630s	4.949s	2.609s	1.439s	0.849s	0.549s
1024	102.640s	75.312s	37.782s	19.019s	9.639s	4.955s	2.609s	1.439s	0.839s
2048	205.032s	150.400s	75.328s	37.788s	19.028s	9.649s	4.959s	2.609s	1.429s
3072	307.456s	225.480s	112.864s	56.566s	28.417s	14.340s	7.299s	3.789s	2.019s
4096	409.856s	300.560s	150.408s	75.332s	37.800s	19.036s	9.648s	4.959s	2.609s

Processing times for the CPU 948

The processing times are subject to the following conditions:

- two CPU 948s are connected
- The active CPU has parameters assigned for for data transfer with the "open driver" in mode 3 (fixed user data length, unsymmetrical). In the static parameter set a character delay time of 220 ms is selected (even parity, 8 bits per character, 1 stop bit).
- The passive CPU has parameters assigned for for data transfer with the "open driver" in mode 1 (fixed user data length). In the static parameter set a character delay time of 220 ms is selected (even parity, 8 bits per character, 1 stop bit).

The processing time is the time
- from starting the job on the active CPU 948 (the user program sets SCB bit 7)
- until the job is completed on the active CPU 948 (the system program resets SCB bit 7)

SEND job

The times were measured
- with different transmission speeds and
- with various amounts of user data

No. of bytes	Transmission speed								
	110	150	300	600	1200	2400	4800	9600	19200
1	0.487s	0.427s	0.347s	0.307s	0.287s	0.277s	0.277s	0.279s	0.279s
2	0.582s	0.499s	0.379s	0.320s	0.299s	0.279s	0.279s	0.279s	0.279s
4	0.782s	0.639s	0.450s	0.359s	0.309s	0.289s	0.279s	0.279s	0.279s
8	1.182s	0.939s	0.599s	0.437s	0.349s	0.309s	0.289s	0.285s	0.279s
16	1.192s	1.519s	0.899s	0.579s	0.419s	0.349s	0.309s	0.289s	0.289s
32	3.583s	2.699s	1.479s	0.874s	0.569s	0.419s	0.349s	0.309s	0.295s
64	6.783s	5.039s	2.657s	1.459s	0.860s	0.569s	0.419s	0.349s	0.309s
128	13.183s	9.739s	4.999s	2.633s	1.449s	0.859s	0.569s	0.419s	0.349s
256	25.985s	19.120s	9.698s	4.979s	2.624s	1.449s	0.859s	0.569s	0.419s
384	38.787s	28.510s	14.389s	7.329s	3.799s	2.029s	1.159s	0.719s	0.499s
512	51.589s	37.899s	19.081s	9.679s	4.969s	2.619s	1.449s	0.859s	0.569s
1024	102.797s	75.449s	37.859s	19.060s	9.669s	4.969s	2.619s	1.449s	0.869s
2048	205.212s	150.548s	75.410s	37.839s	19.059s	9.660s	4.969s	2.629s	1.458s
3072	307.628s	225.624s	112.960s	56.619s	28.449s	14.359s	7.319s	3.799s	2.040s
4096	409.043s	300.742s	150.510s	75.390s	37.839s	19.059s	9.669s	4.979s	2.629s

6.5 Mode 4 - Printer Output

6.5.1 Parameter Assignment

The following sections explain how you can assign parameters for data transmission with the "open driver" in mode 4 and how you should structure the required data blocks.

You have two options:

- either assign parameters directly in STEP 5 by inputting them into data blocks
- or enter the parameters via the menus in the COM PP Parameter Assignment Software and then transfer the data blocks you have created to the CPU using the STEP 5 basic package. Working with COM PP is described in Chapter 3.

Assigning parameters to DX 2

DX 2 contains the link type. It also contains the pointers to the required parameters and parameter sets. The following table shows the values you can use when assigning parameters to DX 2. The assignment in DX 2 begins at DW 0 (absolute). All the numerical values are in hex.

DW	Parameter	Significance
0	4D41	MA
1	534B	SK
2	5832	X2
Link type		
3	0030	Data transmission with the "open driver"
Pointer to static parameter set		
4	44xx or 58xx	DB no. xx or[2)] DX no. xx
5	xxxx	from DW no. xxxx
-		
6	0000	reserved
7	0000	reserved
Pointer to transmit mailbox		
8	44xx or 58xx	DB no. xx or [2)] DX no. xx
9	xxxx	from DW no. xxxx
10	xxxx	Length of transmit mailbox in words (range of values 0002H to 0801H)
-		
11	0000	reserved
12	0000	reserved
13	0000	reserved
Pointer to SCB/RCB [1)]		
14	4D00 or 5300 or 44xx or 58xx	Flag or [2)] S flag or DB no. xx or DX no. xx
15	xxxx	FW no., SW no. or DW no. xxxx

1) The locations of SCB and RCB must be selected together. For a detailed description, refer to Section 6.5.1.
2) Range of values for xx from 3 to FF (3 - 255)

Example

In this example you can see how to assign the following basic parameters to a DX 2:

- The static parameter set is in data block 10 from data word 4 onwards.
- The transmit mailbox is in data block 6 from data word 2 onwards, length 10 words.
- The send and receive coordination bytes (SCB and RCB) are in flag word 6.

With these parameters, DX 2 appears as below:

DX 2

0 :	KH = 4D41;	
1 :	KH = 534B;	'MASKX2'
2 :	KH = 5832;	
3 :	KH = 0030;	**LINK TYPE** "open driver"
4 :	KH = 440A;	**STATIC PARAMETER SET** IN DB 10
5 :	KH = 0004;	FROM DATA WORD 4
6 :	KH = 0000;	reserved
7 :	KH = 0000;	
8 :	KH = 4406;	**TRANSMIT MAILBOX** IN DATA BLOCK 6
9 :	KH = 0002;	FROM DATA WORD 2
10:	KH = 000A;	LENGTH 10 WORDS
11:	KH = 0000;	
12:	KH = 0000;	reserved
13:	KH = 0000;	
14:	KH = 4D00;	**COORDINATION BYTES** IN FLAG AREA
15:	KH = 0006;	SCB FLAG BYTE 6, RCB FLAG BYTE 7
16:	KH = xxxx;	no longer belongs to the parameter assignment for DX 2, content irrelevant, does not need to exist
:		
:		

Static parameter set

In the static parameter set, you must enter the parameters for the physical layer as well as specific transmission parameters.

The following table shows the values that you can enter in a data block or extended data block for the static parameter set. The parameter assignment begins at the DW specified in DX 2.

All numerical values are shown in hexadecimal format.

DW (rel.)	Range of values	Significance
Transmission speed		
0	0001 0002 0003 0004 0005 0006 0007 0008 0009	110 bps 150 bps 300 bps 600 bps 1200 bps 2400 bps 4800 bps 9600 bps 19200 bps (only with V.25 submodule and RS 422-A/485 submodule)
Parity		
1	0000 0001 0002	no odd even
Bits per character		
2	0006 0007 0008	6 bits per character 7 bits per character 8 bits per character
Stop bits		
3	0001 0002 0003	1 stop bit 1 + 1/2 stop bits 2 stop bits
Flow control		
4	0000 0001 0002	none XON/XOFF RTS/CTS
Mode		
5	0004	printer output
-		
6	0000	reserved
7	0000	reserved
8	0000	reserved
9	0000	reserved
10	0000	reserved

DW (rel.)	Range of values	Significance
End identifier		
11	0000 00xx	00 (NUL) code of the first end identifier no second end identifier xx code of the first end identifier (range of values 01 - FF)
12	0000 00xx	no second end identifier xx code of the second end identifier (range of values 01 - FF)
-		
13	0000	reserved
Printer output		
14	0000 to 00FF	Wait time after sending CR (carriage return) (unit: 10 ms)
15	0000 to 00FF	Wait time after sending LF (line feed) (unit: 10 ms)
16	0000 to 00FF	Wait time after sending FF (form feed) (unit: 10 ms)
17	0001 to 00FF	Lines per page (including header and footer lines)
18	0000 to 00FF	Left margin (number of blanks)
19	00xx	Specifies a token character xx in the header or footer for the page number (range of values 20 - 7F)
20	0000 0001 0002	Number of header lines
21	0000 0001 0002	Number of footer lines
22	 0001 0002 0003	Delimiter or line termination character CR LF CR LF
23ff		Header and footer lines including the delimiter from rel. DW 22 (max. 554 characters)

Meaning of the parameters

Transmission speed

Specified in bps.

Parity

Number of bits with the value "1" in a string of information bits with a uniform length (e.g. 1 byte). Depending on the value, the parity is "even" or "odd". To check the parity, the string of information bits is extended by a further bit, the parity bit ("0" or "1") which when added to the information bits produces a selectable parity state.

Bits per character

Number of data bits used to form a character.

Stop bits

Duration of the stop bits, relative to the time required to transmit an information bit. The stop bits follow each character transmitted in a start-stop transmission.

Flow control

A mechanism with which the transmission (CPU or partner) and reception (partner or CPU) of data is synchronized when the data source operates faster than the data sink.
The control of the data flow using flow control is transparent for the user (with the exception of the required parameter assignment). This means that if you use flow control 0001 (XON/XOFF, control characters DC1 and DC3), the user data must not contain these control characters.
If you use flow control 0002 (RTS/CTS) you must also wire signals RTS and CTS (see also Section 5.8 of the S5-135U/155U System Manual).

Mode

Version of the possible methods of data transmission.

Mode 4 (printer output)

Sending

When you output on a printer, the transmit mailbox up to and including the end identifier (as programmed in relative DW 11 and DW 12) is transmitted. In the first word of the transmit mailbox, you must specify the length (in bytes) of the user data relevant for the transmission. The end identifier is included in the user data. This specified length does not include the word containing the length specification itself. The length specification is not transmitted. Words 14 to 23 ff. of the static parameter set are only relevant in this mode.

During the output, the data are prepared for printing (parameters from address 14H in the specific static parameter set). The print job is terminated after the end identifiers have been sent, the line counter and similar values required for printing remain valid.

Receiving

Reception of characters does not take place in this mode.The only exception to this are the control characters for flow control if this has been selected (flow control 1). Any characters that may be received are discarded.

End identifier

The data transmission is terminated by the specified end identifier. In the first end identifier, the entry NUL is a valid end character. This ensures that as standard, at least one valid end identifier is assigned. The second end identifier is optional and is only effective when the first end identifier is not NUL. In the second end identifier the entry NUL is counted as a non-assigned end character. If both end identifiers are specified (both <> NUL), then both together define the end of the message.

Wait times

In the static parameters, you can select wait times for the output of carriage return (CR), line feed (LF) and form feed (FF). No further characters are output during the selected wait time.

CR: control character "carriage return"
LF: control character "line feed"
FF: control character "form feed", in the protocol, carriage return and paper transport to the start of the next page are executed.

Lines per page

The number of lines output is recognized based on the number of output delimiters (in rel. DW 22). The number of lines per page also includes the header and footer lines. The value of the lines per page parameter must therefore be at least one higher than the total number of header lines in DW 20 (relative) and the number of footer lines in DW 21 (relative):

$LpP \geq (No.\ HL + No.\ FL) + 1$

Left margin

The left margin precedes every text, header or footer line in the form of blanks. You must make sure that the total length of a line can be handled by the printer.

Token character for page number

The page number is printed out with a maximum of four digits and can be selected in the header or footer. The header or footer line must contain the character defined in the static parameter DW 19 (relative) as the token character. During output, the driver replaces this character by the page number with leading zeros. The highest page number that can be output depends on the number of token characters. If the actual page number exceeds this value, the page numbering begins again at 1. The highest page number is therefore 0000!

Number of footer and header lines

This value specifies the number of lines required for the header or footer. Remember that the number of lines per page also includes the header or footer lines.

Delimiters

The defined delimiter (DW 22 relative) for delimiting text lines as well as header and footer lines must be contained in the text, header and footer to be output.

Caution

When you assign the end identifier(s) remember that if you operate with 6 or 7 bits per character, bits 6 and 7 or bit 7 of the selected end identifier(s) are not transmitted.

Note

The driver does not insert printer control characters (CR, LF, FF) between individual print jobs. These characters must be contained in the text.

The value entered in the static parameter set DW 17 for "lines per page" has no effect on the set page length ("lines per page") of the printer connected.

The control character FF (form feed) contained in the text is only output as the control character for form feed to the printer within the header and footer lines. Within the text the control character FF (form feed) is converted to delimiters. The number of delimiters output depends on the parameter "lines per page" and the internal line counter.

Example:

The example illustrates how to write a static parameter set in DB 10 from DW 4 onwards for printer output.

DB 10

0:	KH = xxxx;	does not belong to the static parameter set,
1:	KH = xxxx;	content irrelevant
2:	KH = xxxx;	
3:	KH = xxxx;	
4:	KH = 0008;	**transmission rate** = 9600 bps
5:	KH = 0002;	**parity** = 2 = even
6:	KH = 0007;	**bits per character** = 7
7:	KH = 0001;	**stop bits** = 1
8:	KH = 0001;	**flow control** = XON/XOFF
9:	KH = 0004;	**mode** = 4 = printer output
10:	KH = 0000;	reserved
11:	KH = 0000;	character delay time is not required in mode 4
12:	KH = 0000;	
13:	KH = 0000;	reserved
14:	KH = 0000;	
15:	KH = 0010;	**code of the first end identifier** = 10H (DLE)
16:	KH = 0004;	**code of the second end identifier** = 04H (EOT)
17:	KH = 0000;	fixed message length is not required
18:	KH = 0032;	**wait time after sending CR** = 50 x 0.01sec (= 0.5sec)
19:	KH = 0032;	**wait time after sending LF** = 50 x 0.01sec (= 0.5sec)
20:	KH = 0064;	**wait time after sending FF** = 100 x 0.01sec (= 1sec)
21:	KH = 0042;	**lines per page** = 66
22:	KH = 0008;	**left margin** = 8 blanks
23:	KH = 0023;	**token character for page number** = "#"
24:	KH = 0002;	**header lines** = 2 lines
25:	KH = 0001;	**footer lines** = 1 line
26:	KH = 0003;	delimiter = 3 = CR LF
27:	KH = 070C;	1st header line line with control character:
28:	KS ='Siemens Simatic S5 ';	07 - BEL (bell character)
40:	KS =' Date: 01.06.91';	0C - FF (form feed)
52:	KH = 0D0A;	delimiter: 0D 0A - CR LF
53:	KH = 1B30;	2nd header line with control character:
54:	KS ='application example CPU ';	1B 30 - ESC 0 (underlining on)
66:	KS = '928B - PT89; Page: ##';	1B 39 - ESC 9 (underlining off)
78:	KH = 1B39;	
79:	KH = 0D0A;	delimiter: 0D 0A - CR LF
80:	KS ='Copyright Siemens AG 199';	1st footer line
92:	KS ='1 ';	
93:	KH = 0D0A;	delimiter: 0D 0A - CR LF
94:	KH = xxxx;	is no longer required, content irrelevant,
:		does not need to exist
:		

Note

For data transmission with the "open driver" the parameters of the physical layer on the communications partner must match those of the CPU 928B/CPU 948.

Transmit mailbox

The transmit mailbox is located in a data block (DB) or extended data block (DX).

In the the transmit mailbox you specify the number of data to be sent (in bytes) as well as the length specification in the first word (in bytes). The specified length does not include the word containing the length specification itself. After this come the data to be transmitted and the end identifier specified in the static parameter set. The end identifier must be included in the data specified in the first word. The data in the transmit mailbox up to and including the assigned end identifier are transmitted. The length specified in the first data word is not transmitted.

The length of the data to be transmitted must not exceed the length of the transmit mailbox minus one word (see above). The length of the transmit mailbox is a maximum of 2049 words (4098 bytes). You specify this in DX 2.

Remember to specify the length of the transmit mailbox in **words** in DX2. but the length of the data to be transmitted in **bytes** in the first word of the transmit mailbox. Select the length of the transmit mailbox as suitable for your particular application.

Example: transmit mailbox

The example illustrates how to assign parameters for a transmit mailbox in data block 6 from data word 2 onwards with a length of 10 words. The first end identifier 10H and the second end identifier 04H were selected in the static parameter set. The number of useful bytes relevant for transmission in this example is 12.

DB 6

0:	KH = xxxx;	does not belong to transmit mailbox, content irrelevant
1:	KH = xxxx;	
2:	KH = 0009;	start of transmit mailbox, length in bytes
3:	KH = 0102;	data to be transmitted: 0102
4:	KH = 0304;	data to be transmitted: 0304
5:	KH = 0506;	data to be transmitted: 0506
6:	KH = 0710;	data to be transmitted: 07, first end id: 10
7:	KH = 04zz;	second end id: 04, zz is not transmitted
8:	KH = zzzz;	relevant user data, not transmitted here
9:	KH = yyyy;	the content of the transmit mailbox is no longer relevant and
10:	KH = yyyy;	not transmitted
11:	KH = yyyy;	
12:	KH = xxxx;	
:		no longer belongs to the transmit mailbox, content
:		irrelevant, does not need to exist

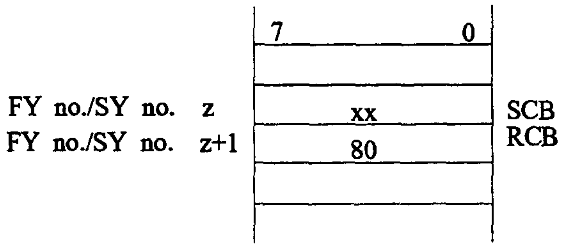
Send and receive coordination bytes (SCB and RCB)

With the SCB you can start the following jobs:

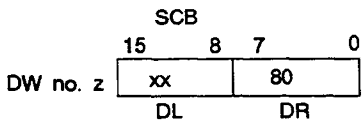
- SEND job
- reset the "open driver"

The coordination bytes can be flags or S flags or be located in a data block or extended data block. Flags and S flags already exist on the CPU 928B or CPU 948, you must set up data blocks and extended data blocks yourself.

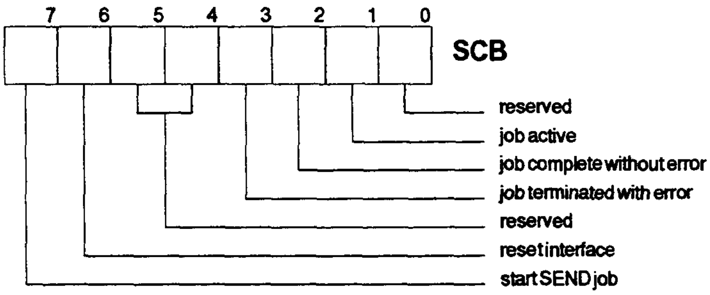
Representation in the flag or S flag area



Representation in a data block or extended data block



Send coordination byte (SCB) The send coordination byte is structured as follows:



The individual bits or bit groups, providing they are not reserved, are described below.

Bit 1 "Job active"

- set by the system program when a job starts
- reset by the system program when a job is completed
- you can read this bit, to know whether there is any point in starting a new job

Bit 2 "Job complete without error"

- set by the system program when a job is completed without error
- you can read this bit and then reset it, or it is reset by the system program when bit 1 "job active" is set for a new job
- you can read this bit, to know whether a job was completed without error

Bit 3 "Job terminated with error"

- set by the system program when a job is terminated with an error
- you can read this bit and then reset it, or it is reset by the system program when bit 1 "job active" is set for a new job
- you can read this bit, to know whether a job was completed with an error

Bit 6 "Reset interface"

- you can set this bit to reset the interface
- reset by the system program as soon as the system program resets the interface

Bit 7 "Start SEND job"

- you can set this bit to start a SEND job
- reset by the system program as soon as the system program completes the SEND job

When you reset the interface, the parameters of the data link layer in the static parameter set (from DW 5 onwards) are read in again (see Section 6.5.1): mode, character delay time, parameters for printer output.

If the SCB bits 6 and 7 are sent simultaneously or if SCB bit 6 is set first followed by SCB bit 7 at an interval ≤ 10 ms (CPU 928B) or ≤ 20 ms (CPU 948), the system program first carries out the reset followed by the SEND job. If SCB bit 7 is set first followed by SCB bit 6 at an interval ≤ 10 ms (CPU 928B) or ≤ 20 ms (CPU 948), the sequence of the two jobs cannot be guaranteed. By resetting, you abort the currently active partner SEND job.

Note

You can only modify the parameters of the physical layer involving the physical transmission characteristics and the flow control (static parameter set up to and including DW 4) using a cold restart on the CPU 928B or on the CPU 948 using a cold restart or power up. Resetting the interface is not sufficient.

Receive coordination byte (RCB)

The receive coordination byte (RCB) is irrelevant for printer output.

Caution

During a cold restart and following power up, 00H is written to the SCB (no job active) and 80H to the RCB (receive mailbox free).

6.5.2 Getting Started

Preparations

You have already installed the interface submodule in the CPU 928B or in the CPU 948 (see Chapter 2), the CPU is plugged into the PLC rack and the power supply is off.

Procedure

We recommend the following procedure for starting data transmission with the "open driver":

Step	Action
1	Connect the CPU 982B/CPU 948 to the partner (with a standard connecting cable).
2	Switch on the power supply.
3	Transfer your user program to the CPU 928B. The user program must contain the following **data blocks** or **extended data blocks** to be able to set the SI 2 interface for data transmission with the "open driver". DX 2 DB/DX with the static parameter set DB/DX with the transmit mailbox [1] DB/DX with the SCB and RCB (only when the SCB/RCB are located in a DB/DX) [2]
4	Perform a **cold restart**.
5	Prepare the partner for data transmission with the "open driver".

[1] Only when the CPU 928B/CPU 948 triggers active jobs itself
[2] If the SCB and RCB are flags or S flag words, they already exist in the CPU 928B/CPU 948 and do not need to be transferred.

6

CPU 928B

During a cold restart or following power up the system program checks that the following exist:

- DX 2,
- the static parameter set
- SCB and RCB

and the correct parameter assignment in DX 2

DX 2 exists and has correct parameters

- The system program initializes the SI 2 interface for data transmission with the "open driver".
- The red LED SI 2 goes off as soon as data transmission with the "open driver" is ready for operation.
- The system program resets the SCB bit 7 (i.e. no SEND job active).
- The system program sets the SCB bit 7 (i.e. receive mailbox enabled).

↓

Data transmission with the "open driver" is ready for operation

DX 2 does not exist or does not have correct parameters

- The CPU remains in the STOP mode.
- The red LED SI 2 is lit in addition to the STOP LED.
- The system program outputs an error message to RS 3 and RS 4.

CPU 948

During a cold restart or following power up the system program checks that the following exist:

- DX 2
- the static parameter set
- SCB and RCB

and the correct parameter assignment in DX 2

DX 2 exists and has correct parameters

- The system program initializes the SI 2 interface for data transmission with the "open driver".
- The red LED SI 2 goes off as soon as data transmission with the "open driver" is ready for operation.
- The system program resets the SCB bit 7 (i.e. no SEND job active).
- The system program resets the RCB bit 7 (i.e. receive mailbox enabled).

↓

Data transmission with the "open driver" is ready for operation

DX 2 does not exist or does not have correct parameters

- The system program does not initialize the SI 2 interface for data transmission with the "open driver"
- The red LED SI 2 does not go off

6.5.3 Operation

Preparations

The SI 2 interface has parameters assigned for data transmission with the "open driver". The SI 2 LED is off. Data transmission with the "open driver" is ready for operation.

Method

The system program now checks constantly whether the send coordination byte (SCB) and receive coordination byte (RCB) exist. If they do not exist (e.g. because the DB/DX has been deleted), the system program calls OB 35. The error number is entered in ACCU 1 (see Chapter 11).

The system program now processes the jobs you have programmed in your user program, or the jobs triggered by the communications partner. The following sections explain how to program the jobs in your user program.

SEND job

If you want to program a SEND job, remember the following sequence for the various modes:

Start job

Check SCB bit 7

Bit 7 is set, a job is active

The transmit mailbox and the SCB must not be modified

Wait until this job is complete

Bit 7 is reset, no job is active

Copy the data to be sent to the transmit mailbox after the first data word

Enter the length of the data to be sent in the first data word of the transmit mailbox

Mark the end of the useful data within the relevant useful data byte in the transmit mailbox with the end identifier(s) defined in the static parameter set

Start the SEND job by setting the SCB bit 7

The system program now processes your job

6

Note

The editing of the print data with header and footer lines is carried out automatically without further activity by the user program.

A receive coordination byte (RCB) is not required. This byte has the value 80H written to it following power up and during a cold restart.

The following diagram illustrates the reactions that can occur if the SCB bit 7 is set and the system program is processing your job.

SCB bit 7 is set

- The transmit mailbox exists
  - The system program transmits the data including the end identifier
    - The job was completed without errors
      - The system program
        - transmitted the data from the transmit mailbox to the partner
        - reset the SCB bit 7
    - The job was completed with an error
      - The system program
        - called OB 35: ACCU 1 contains the error number
        - reset the SCB bit 7
- The transmit mailbox does not exist
  - The system program
    - did not process the job
    - called OB 35 and entered an error number in ACCU 1
    - reset the SCB bit 7

Note

You can only start a new SEND job when the previous job is completed.

Resetting the interface

You have set the SCB bit 6 and have therefore started a reset job. The following diagram shows the possible reactions.

SCB bit 6 is set

- the static parameters from DW 5 onwards are read in and evaluated
- page and line counters are reset
- when the reset job is complete, the SCB bit 6 is reset

6.5.4 Protocol Definition of the "Open Driver" in Mode 4

The following section is intended to help you if you want to connect an existing printer using the "open driver" in mode 4 (printer output).

The following aspects of transmitting and receiving are described:

- structure of the transmitted/received data frame
- any restrictions
- basic sequence of events

The sequence of events refer to the communications processor. The heavy lines indicate the standard sequence, whereas the thinner lines represent error handling.

Transmitting

Setting up the transmit mailbox is described in Section 6.5.1 "Transmit mailbox". The sequence of the SEND job is described in Section 6.5.3 "SEND job".

In mode 4, only the user data are transmitted. The length information contained in the first word of the transmit mailbox is not transmitted.

The data frame to be transmitted is structured as shown below:

Length of the user data: N bytes

1 ... N	N+1	N+2
User data - length N bytes	E1	E2

E1: first end identifier
E2: second end identifier, only exists if assigned in static parameter set

The end identifier selected in the static parameter set must be located in the transmit mailbox after the user data to be sent.

Caution

If you select flow control with XON/XOFF, the user data must not contain XON (DC1 = 11H) or XOFF (DC3 = 13H).

The print editing is executed according to the static parameters (wait times, margin, page counter, delimiters and control characters).

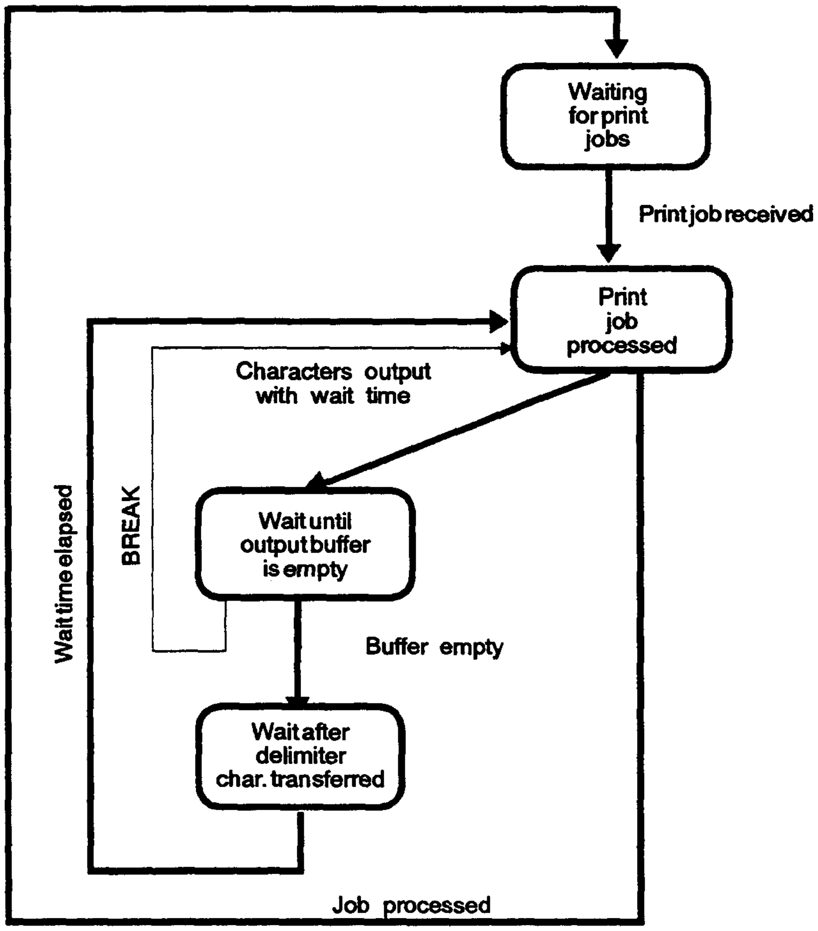


Fig. 6-12 Sequence of transmission in mode 4

Data Transmission with SINEC L1

7

Contents of Chapter 7

7 Data Transmission with SINEC L1

You can use the second serial interface of the CPU 928B/CPU 948 to

- form a connection between the CPU and the SINEC L1 local area network.

The SINEC L1 interface module is required to adapt the second interface to the CPU.

7.1 Introduction

SINEC L1 is a local area network (LAN) for networking groups of SIMATIC S5 programmable controllers and third-party devices. It operates on the master-slave principle.

Master-slave principle

- The **master** is a programmable controller which coordinates and monitors the data traffic within the LAN. The master must be equipped with the CP 530 communications processor.
- The **slave** is any other node (station) within the communications network.
  Every slave has its own node number.

One master and up to 30 slaves can be connected to the SINEC L1 local area network.

Polling list

The master controls the LAN on the basis of a polling list which you generate when you assign the master parameters. The polling list contains the number of all slaves (max. 64 entries) which are involved in data communication. The number and sequence of entries in the polling list define the order in which the nodes (slaves) are addressed within one cycle depends on how often it is specified in the list (i.e. more than once if required).

Interrupt processing

The normal bus cycle can be interrupted by an interrupt request. If a node requests an interrupt, it fills a transmit mailbox with data and sets the interrupt bit (SCB bit 4) together with the "Permission to send" bit (SCB bit 7). From this point onwards, the operating system continously attempts to log this express message with the bus master and thus send the message. As soon as the master recognizes the request, it grants the slave requesting the interrupt an "Interrupt cycle" and then resumes the interrupted bus cycle.

Interrupt list

The interrupt list contains the number of all nodes which may place interrupt requests with the master. The order in which the node numbers are entered in the list defines the processing sequence, i.e. the priority with which interrupts are processed. A maximum of 30 entries may be made in the list; each node may only be entered once.

If you require more information on the principle of operation of the SINEC L1 LAN, please refer to the "SINEC L1 Local Area Network" Manual, order no. 6ES5 998-7LA11 (see Further Reading).

CPU 928B/CPU 948 as slave

Data can be exchanged via the SINEC L1 LAN as follows:

- Between two nodes within the LAN
  Master → Slave (normal communication)
  Slave → Master (normal communication),
  Slave → Slave (internode communication)
- Between one node and all other nodes within the LAN
  Master → All slaves (broadcast),
  Slave → Master and all slaves (broadcast).

The "Normal communication", "Internode communication" and "Broadcast" transmission modes can also be initiated via an interrupt.

A BT 777 bus terminal is required as a level converter for every node (master or slave). Please refer to the "SINEC L1 Local Area Network" Manual, 6ES5 998-7LA11 for further information about the LAN structure (see Further Reading).

CPU 928B/CPU 948 as point-to-point master

Apart from operating as a slave in the LAN, the CPU 928B/CPU 948 can also be used as a point-to-point master and linked to a slave. In this case, the allocation of node numbers is fixed, i.e. 0 for the master and 1 for the slave.

Internode communication, broadcasting and interrupt processing are not meaningful in this configuration since only two nodes are involved in the data exchange.

You can use a drop cable instead of the BT 777 terminal. This cable is described in the S5-135U/155U System Manual, Section 5.8, "Interface Submodules".

User program

SINEC L1 controls the bus and the transmission of data. You can determine how much and which data are transferred from one node to another in your user program. SINEC L1 communications with your program by means of the send (SCB) and receive (RCB) coordination bytes.

7.1.1 Applications

Data transmission with the SINEC L1 involves the exchange of data between the CPU 928B/CPU 948 and one or several partners which are connected to the SINEC L1 local area network.

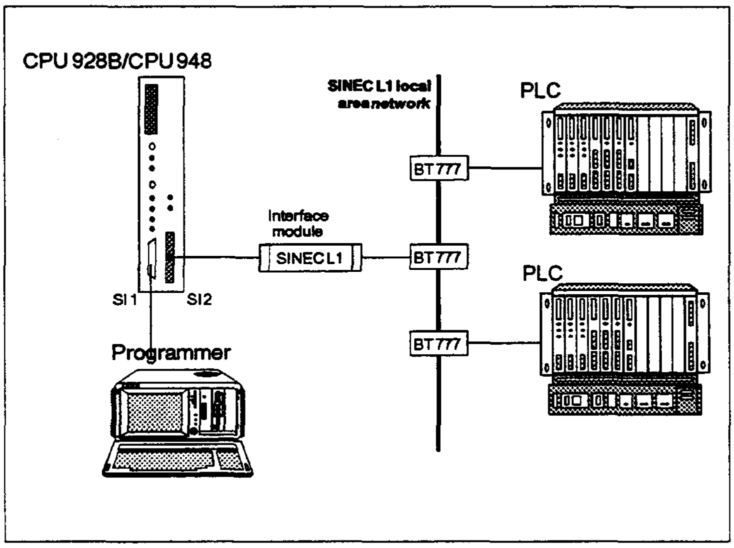


Fig. 7-1 Data exchange

To connect the CPU 928B or the CPU 948 to the SINEC L1, you need a SINEC L1 interface module. Insert this module into the receptable of the second interface of the CPU 928B/CPU 948. Make the connection to the LAN by means of a BT 777 terminal. If the CPU 928B is to operate as a point-to-point master, then you can use a drop cable instead of a transceiver.

Please refer to the S5-135U/155U System Manual (Section 5.8) for further information on the interface module and connector pin assignments.

Transmission speed

The baud rate is 9 600 bps.

Data transmission

The SINEC L1 is capable of

- Sending data
  The data to be sent are stored in the transmit mailbox (see Section 7.2.2).
- Receive data
  The data to be sent are stored in the receive mailbox (see Section 7.2.3).
  A change in operating state can also be initiated when data are received.

7.1.2 Integration into User Program

Preliminaries

Before generating the user program, you should decide the following:

- With which partners is the CPU 928B or the CPU 948 to exchange data?
- Which data are to be exchanged?

Programming

The data to be sent are stored in the data block (DB/DX) defined as the transmit mailbox. You can instruct the operating system to send the contents of this mailbox by means of the send coordination byte (SCB).

The received data are stored in the data block (DB/DX) defined as the receive mailbox. You can enable and disable this mailbox by means of the receive coordination byte (RCB).

Assigning parameters

Data block DX 2 is reserved for assigning the parameters of the second CPU 928B/CPU 948 interface. The location and length of the send and receive mailboxes plus the location of the SCB and RCB are entered in this block.

Note

If there is no DX 2 in the CPU, then the second interface is initialized as a programmer port.

Note

You must define the parameters in DX 2 before you start transmitting data via the SINEC L1.

With the CPU 928B, the parameters in DX 2 are only adopted by the system program following a cold restart.
With the CPU 948, the parameters are also adopted by the system program following power up or an overall reset (loading DX 2 from memory card).

You may only change the contents of the transmit mailbox and the SCB on condition that no data send enabling command has been given.

You may only change the contents of the receive mailbox and the RCB on condition that data reception is disabled.

The following sections contain detailed information about values which can be assigned to SINEC L1 parameters.

7.2 Assigning Parameters

The following sections describe how you can assign parameters to SINEC L1 and how the DX 2 block is structured.

7.2.1 Assigning Parameters to DX 2

The node number and interface type are noted in DX 2. It also includes the pointers to all the required parameters and parameter sets.

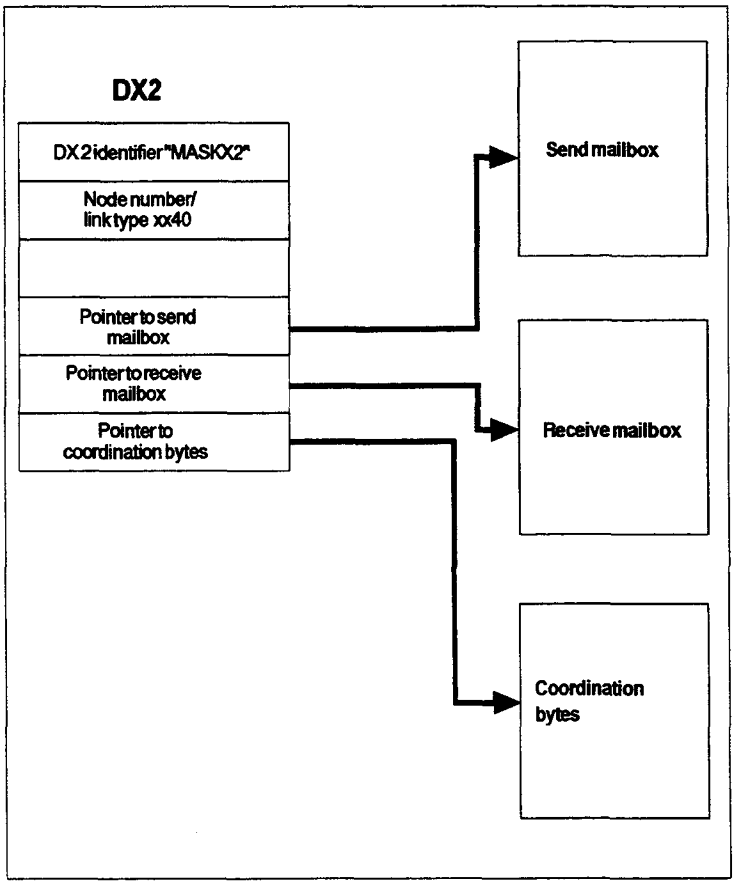


Fig. 7-2 Components of the user program

The values which you can assign parameters to the DX 2 block are listed in the following table. The parameter settings start at DW 0. All numerical values in the "Parameter" column are represented in hexadecimal form.

DW	Parameter	Meaning
0	4D41	DX 2 identifier "MASKX2"
1	534B	
2	5832	
Interface type		
3	xx40	Interface type SINEC L1 Value range for xx: 00 Point-to-point master 01H to 1EH (1 - 30) Slave with node number
4	0000	reserved
5	0000	
6	0000	
7	0000	
Pointer to transmit mailbox		
8	44xx or 58xx	DB no. xx or [1] DX no. xx
9	xxxx	From DW xxxx onwards [2]
10	00xx	Length of transmit mailbox in words [2]
Pointer to receive mailbox		
11	44xx or 58xx	DB no. xx or [1] DX no. xx
12	xxxx	From DW no. xxxx onwards [2]
13	xxxx	Length of receive mailbox in words [2]
Pointer to SCB/RCB [3]		
14	4D00 or 5300 or 44xx or 58xx	Flag or S flag or DB no. xx or [1] DX no. xx
15	xxxx	FW no., SW no. or DW no.

[1] Value range for xx from 03H to FFH (3 to 255)
[2] Value range for xx from 02H to 21H (2 to 33)
[3] The SCB and RCB can only be defined as having a common location.

Example

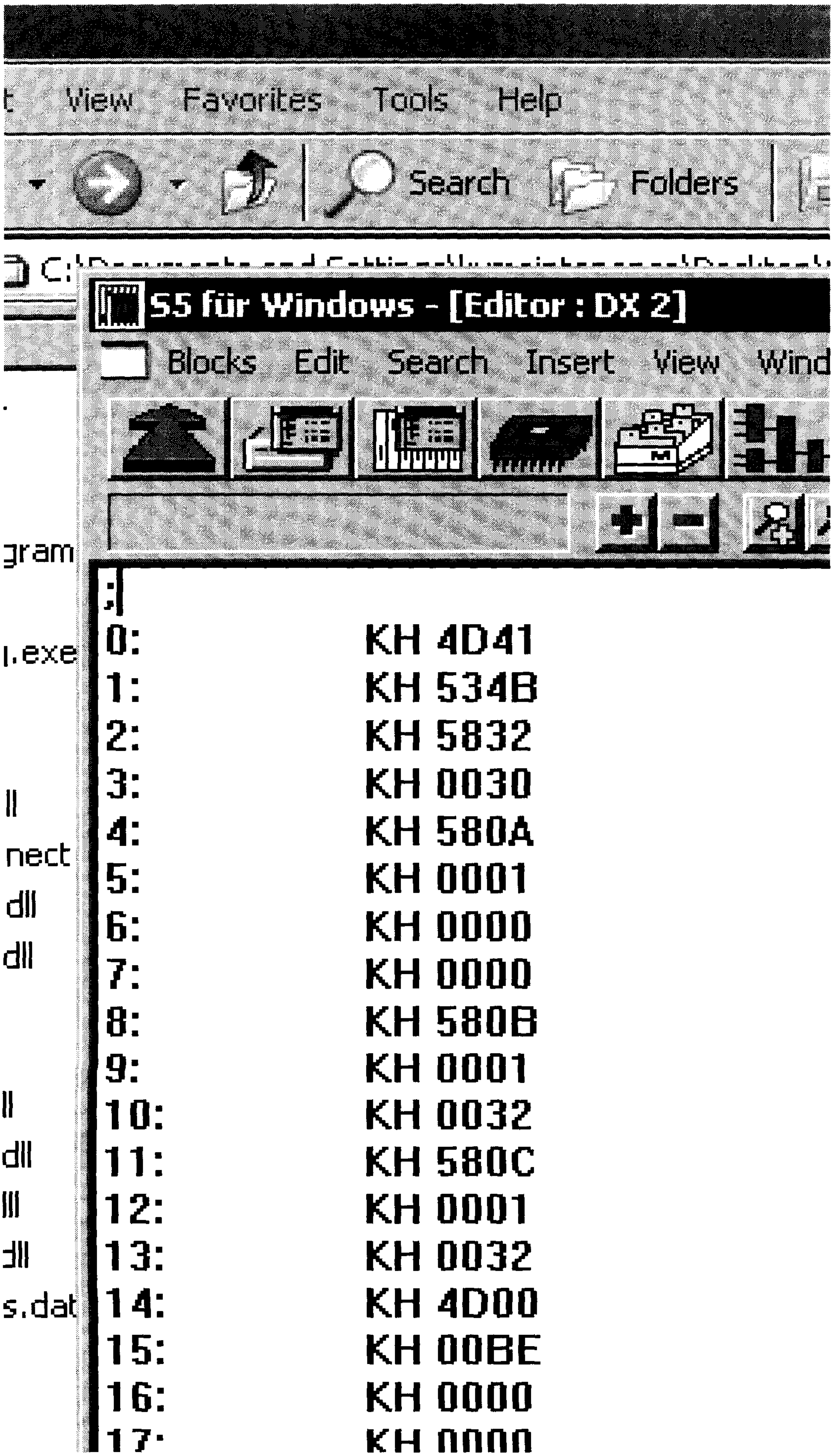
This example shows you how you can assign the following basic data to DX 2:

- The transmit mailbox is located in data block 6 from data word 2 onwards; 10 words in length
- The receive mailbox is located in data block 8 from data word 1 onwards; 6 words in length
- SCB and RCB are located in flag word 6.

The parameter settings are thus as follows:

DX 2

0 : KH = 4D41;	"MASKX2"
1 : KH = 534B;	
2 : KH = 5832;	
3 : KH = 0840;	**INTERFACE TYPE** SINEC L1, NODE NO. 8
4 : KH = 0000;	Reserved
5 : KH = 0000;	Reserved
6 : KH = 0000;	Reserved
7 : KH = 0000;	Reserved
8 : KH = 4406;	**TRANSMIT MAILBOX** IN DB 6
9 : KH = 0002;	FROM DW 2 ONWARDS
10: KH = 000A;	LENGTH 10 WORDS
11: KH = 4408;	**RECEIVE MAILBOX** IN DB 8
12: KH = 0001;	FROM DW 1 ONWARDS
13: KH = 0006;	LENGTH 6 WORDS
14: KH = 4D00;	**COORDINATION BYTES** IN FLAG AREA
15: KH = 0006;	SCB IN FY 6, RCB IN FY 7
16: KH = xxxx;	Is not part of DX 2 parameter setting, contents optional, can be omitted if desired.
:	
:	


View Favorites Tools Help
Search Folders
S5 für Windows - [Editor : DX 2]
Blocks Edit Search Insert View Wind
;
0: KH 4D41
1: KH 534B
2: KH 5832
3: KH 0030
4: KH 580A
5: KH 0001
6: KH 0000
7: KH 0000
8: KH 580B
9: KH 0001
10: KH 0032
11: KH 580C
12: KH 0001
13: KH 0032
14: KH 4D00
15: KH 00BE
16: KH 0000
17: KH 0000

7.2.2 Transmit Mailbox

The transmit mailbox is located in data block (DB) or an expanded data block (DX).

Structure

The structure of the transmit mailbox is as follows:

Absolute DW no.	DB with transmit mailbox		Relative DW no.
DW0			
DW1			
:			
:			
DWn Total length of transmit mailbox specified (in words) in DX 2	Number of bytes to be sent Value range: 01H to 40H (1 to 64)	Destination (node number) Allocation : 0 Master 1 - 30 Slave 31 Broadcast	DW0
	1st data byte	2nd data byte	DW1
	3rd data byte	4th data byte	Number of bytes to be sent
	:	:	
DWn+32	63rd data byte	64th data byte	DW32
:			

The parameters for the total length of the transmit mailbox (in words) including the word for specifying length and node number are set in DX 2.

Enter the number of bytes to be sent in DL0 of the transmit mailbox; enter the node number of the data destination in the SINEC L1 LAN in DR0. Enter the data to be transmitted from DL1 onwards.

Please note that you can only send as many bytes as can be located in the transmit mailbox.

When the CPU 928B is operating as a point-to-point master, the node number of the data destination in the SINEC L1 LAN is irrelevant, i.e. it is always '1'.

Example

The following example shows how you can set parameters for a transmit mailbox in DB 6, starting at data word 2, with a length of 10 words. 7 bytes must be sent to the master.

DB 6

0: KH = xxxx;	Does not belong to transmit mailbox, contents optional
1: KH = xxxx;	
2: KH = 0700;	LENGTH SPECIFIED IN BYTES, NUMBER OF MASTER
3: KH = 0102;	DATA TO BE SENT: 0102
4: KH = 0304;	DATA TO BE SENT: 0304
5: KH = 0506;	DATA TO BE SENT: 0506
6: KH = 07yy;	DATA TO BE SENT: 07, YY WILL NOT BE SENT
7: KH = yyyy;	CONTENTS OF TRANSMIT MAILBOX, WILL NOT BE SENT
8: KH = yyyy;	HERE
9: KH = yyyy;	
10: KH = yyyy;	
11: KH = yyyy;	
12: KH = xxxx;	
:	Does not belong to transmit mailbox, contents optional, can be
:	omitted if desired.

With the above parameter settings, a maximum of 18 bytes can be sent.

7.2.3 Receive Mailbox

The receive box is located in a data block (DB) or an expanded data block (DX).

Structure

The structure of the receive mailbox is as follows:

Absolute DW no.	DB with receive mailbox		Relative DW no.
DW0			
DW1			
:			
:			
DWn Total length of receive mailbox specified (in words) in DX 2	Number of received bytes Value range: 01H to 40H (1 to 64)	Source (node number) Allocation: 0 Master 1 - 30 Slave	DW0
	1st data byte	2nd data byte	DW1 Number of received bytes
	3rd data byte	4th data byte	
	:	:	
DWn+32	63rd data byte	64th data byte	DW32
:			

The parameters for the total length of the receive mailbox (in words) including the word for specifying length and node number are set in DX 2.

The receive mailbox contains the number of received bytes in DL0 and the node number of the data source in the SINEC L1 LAN in DR0. These values are entered by the system program so that you can see how many data have been received. The received data are entered from DL1 onwards

Please note that you can only receive as much data as can be located in the receive mailbox.

Example

The example below shows the structure of a receive mailbox in DB 8, starting at data word 1, with a length of 6 words. 7 bytes have been received from a slave with node number 3 in internode communication mode.

DB 8

0: KH = xxxx;	Does not belong to receive mailbox.
1: KH = 0703;	LENGTH IN BYTES, SOURCE NODE NUMBER 3
2: KH = 0102;	RECEIVED DATA: 0102
3: KH = 0304;	RECEIVED DATA: 0304
4: KH = 0506;	RECEIVED DATA: 0506
5: KH = 07yy;	RECEIVED DATA: 07, YY NOT RECEIVED
6: KH = yyyy;	CONTENTS OF RECEIVE MAILBOX, NOT RECEIVED
7: KH = xxxx;	Does not belong to receive mailbox, contents optional, can be omitted if desired.
:	
:	

With the parameter settings above, a maximum of 10 bytes can be received.

7.2.4 Coordination Bytes

SCB

You can use the SCB to instruct the operating system to send the contents of the transmit mailbox.

RCB

You can use the RCB to enable and disable the receive mailbox.

The coordination bytes can be located in the flag or S flag area or in a data block or expanded data block.

Representation in flag or S flag area

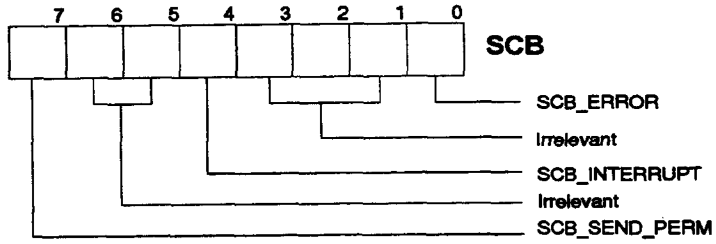
	7 0	
FY No./SY No. z	xx	SCB
FY No./SY No. z+1	yy	RCB

Representation in data block or expanded data block

	SCB	RCB
	15 8	7 0
DW No. z	xx	yy
	DL	DR

SCB

The SCB is structured as follows:



The individual bits which are relevant are described below.

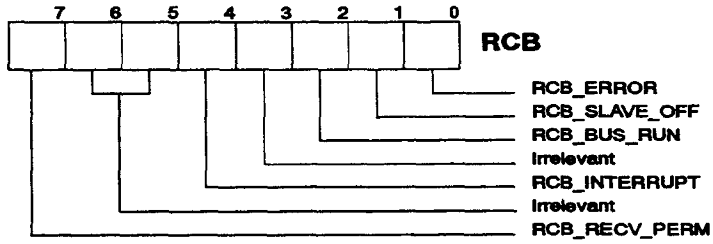
Set by	Reset by	Evaluate by	Reason/Objective
Bit 0 "SCB_Error" [1]			
System program			When a job has been terminated with an error (e.g. no transmit mailbox, negative acknowledgement by receiver, etc.) or the bus monitoring time (250ms) has expired with SCB bit 7 set to 1.
	System program		When a job has been terminated without error.
		User program	To repeat data transmission without errors.
Bit 4 "SCB_Interrupt" [2]			
User program			To insert an interrupt cycle, i.e. to send data to receivers outside the polling list. This bit is valid only if SCB bit 7 is set as well.
	System program		When a job (send data or send data with interrupt) is fully terminated with or without error or the bus monitoring time (250ms) has expired.
		System program	To carry out interrupt processing.
Bit 7 "SCB_SEND_PERM"			
User program			If you want to send data.
	System program		When the transmission of data or an interrupt cycle is complete or the bus monitoring time (250ms) has expired.
		System program	To determine whether data must be sent.
		User program	To determine whether data may be written to the transmit mailbox (SCB bit 7 = 0).

1) The "SCB_ERROR" bit is valid and can be evaluated only if you have given permission to send (SCB bit 7 = 1) and the system program has ended the job (SCB bit 7 = 0).

2) Bit 4 "SCB_INTERRUPT" is not relevant in operation as a point-to-point master. It is sent to 1 every time data have been sent.

RCB

The RCB is structured as follows:



The individual bits or bit groups which are relevant are described below:

Set by	Reset by	Evaluate by	Reason/Objective
Bit 0 "RCB_Error" 1)			
System program			When a receive operation has not been executed correctly.
	System program		When a receive operation has been terminated without error.
		User program	To evaluate reception.
Bit 1 "RCB_SLAVE_OFF" 1)			
System program			When at least 1 slave has failed.
	System program		When all slaves are ready for operation.
		User program	To determine whether all slaves are ready for operation.
Bit 2 "RCB_BUS_RUN" 1)			
System program			When a connection is being established between the master and this CPU 928B.
	System program		When the SINEC L1 LAN is in the STOP state or the bus monitoring time (250ms) has expired.
		User program	To determine whether the bus is ready for operation.
Bit 4 "RCB_INTERRUPT" 2)			
System program			When the received data have been sent by interrupt.
	System program		When the received data have been sent without interrupt.
		User program	To determine whether the received data have been sent by interrupt.

1) These bits are updated in every bus cycle, regardless of whether or not receive permission has been set in RCB bit 7.

2) This bit is valid and can be evaluated only if you have given permission to receive (RCB bit 7 = 1) and the system program has cancelled the permission again (RCB bit 7 = 0).

Bit 7 "RCB_RECV_PERM"			
System program			When the interface is initialized, during cold restart and/or power recovery.
User program			When you enable the receive mailbox.
	System program		When all the received data have been entered in the receive mailbox.
		System program	To determine whether the receive mailbox is enabled.
		User program	To determine whether data are stored in the receive mailbox.

Caution

Make sure that your user program processes the received data as quickly as possible and that the receive mailbox is enabled again immediately afterwards by means of RCB bit 7.

No further data can be received as long as the receive mailbox is disabled by RCB bit 7. The partner receives a negative acknowledgement via the bus.

7.3 Start-Up

Preparations

You have installed the interface module in the CPU 928B or the CPU 948 (see S5-135U/155U System Manual), the CPU is inserted in the CC rack and the voltage is disconnected.

Recommended procedure

We recommend the following procedure for starting up data transmission with the SINEC L1 LAN:

Step	Action
1	Connect the CPU 928B/CPU 948 to the SINEC L1 local area network.
2	Switch on the power supply.
3	Transfer your user program to the CPU 928B. The user program must contain the following data blocks or expanded data blocks which assign parameters to the interface SI 2 for SINEC L1 data transmission: • DX 2 • DB/DX with send mailbox • DB/DX with receive mailbox • DB/DX with SCB and RCB (only when SCB/RCB are in DB/DX) [1)]
4	Cold start the system.

1) If the SCB or RCB is a flag or S flag, it already exists in the CPU 928B/CPU 948 and does not need to be transferred.

CPU 928B

During a cold restart or power recovery, the system program checks whether the following are present:

- DX 2
- SCB and RCB

and the correct assignment of DX 2 parameters

Parameters present and correctly set

- The system program initializes the SI 2 interface for SINEC L1
- The red LED SI 2 goes out as soon as SINEC L1 is ready
- The system program resets SCB bit 7 (send mailbox enabled)
- The system program resets RCB bit 7 (receive mailbox enabled)

SINEC L1 is ready for operation

Parameters not present or not correctly set

- The CPU stays in the STOP state
- The red LED SI 2 lights up in addition to the STOP LED
- The system program outputs an error message in RS 3 and RS 4

CPU 948

During a cold restart or power recovery, the system program checks whether the following are present:

- DX 2
- SCB and RCB

and the correct assignment of DX 2 parameters

Parameters present and correctly set

- The system program initializes the SI 2 interface for SINEC L1
- The red LED SI 2 goes out as soon as SINEC L1 is ready
- The system program resets SCB bit 7 (send mailbox enabled)
- The system program resets RCB bit 7 (receive mailbox enabled)

SINEC L1 is ready for operation

Parameters not present or not correctly set

- The system program does not initialize the SI 2 interface for SINEC L1
- The red LED SI 2 does not go out

7.4 Operation

Preparations

Interface SI 2 is assigned parameters for SINEC L1. LED SI 2 does not light up. SINEC L1 is ready for operation.

Mode of operation

The system program now continually checks whether SCB and RCB are present. If they are not (e.g. because the DB/DX containing the RCB/SCB has been erased), the system program calls up the OB 35. The error code is entered in ACCU 1 (see Chapter 11, "Error Messages").

The section below shows you how to program data transmission in your user program.

7.4.1 Send Data

If you want to send data, the user program should run as follows:

Check SCB bit 7

Bit 7 is set, send mailbox is occupied

The send mailbox and SCB must not be altered

Wait until this job has been completed

Bit 7 is reset, no data are being sent at present, the send mailbox is free

Copy the data to be sent into the send mailbox

Enter the number of bytes to be sent into the mailbox (DL0)

Enter the destination node number on the SINEC LAN into the send mailbox (DR0)

Enable the send mailbox by setting SCB bit 7

The system program will now process your job

The following diagram shows the kind of reactions which may occur when SCB bit 7 is set and the system program is processing your job.

SCB bit 7 is set

The send mailbox is present and correctly set

The system program accepts the data and attempts to transfer them according to the SINEC L1 protocol when next addressing by the master

The job has been terminated without error

The system program has

- transferred the data
- teset SCB bits 4 and 7

The job has been terminated with an error

The system program has

- called up OB 35: ACCU 1 contains the error number
- set SCB bit 0
- reset SCB bits 4 and 7

The send mailbox is not present or not correctly set

The system program has

- not processed the order
- called up OB 35 and entered an error code in ACCU 1
- set SCB bit 0
- reset SCB bits 4 and 7

Note

You cannot start to send data again until the previous send operation has been completed.

7.4.2 Receive Data

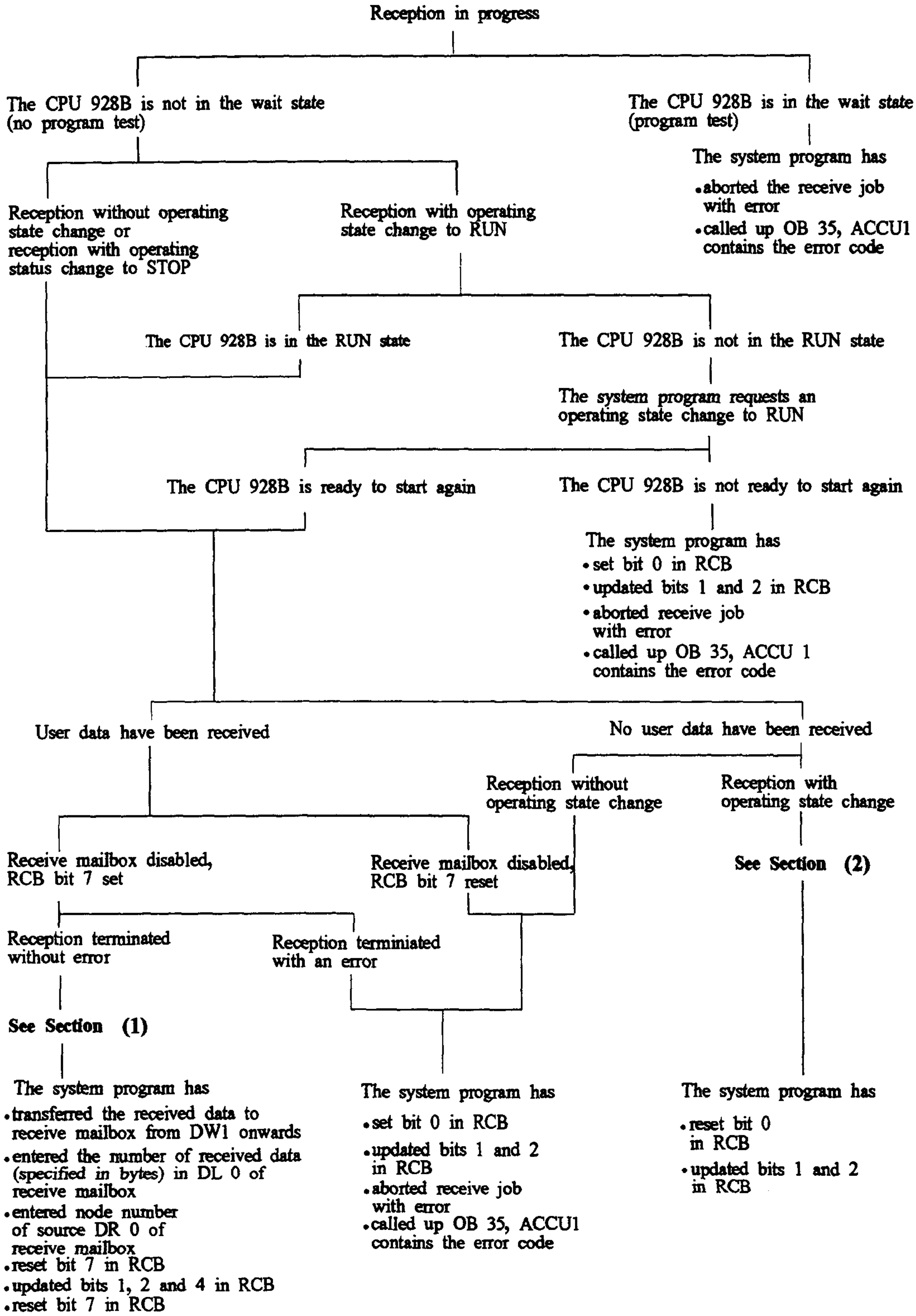
The reception of data is managed as an independent background process by the system program and signalled to the user program via the RCB. The system program resets RCB bit 7 (receive mailbox disabled) when all data have been entered in the receive mailbox. You can now evaluate the received data. The receive mailbox cannot be overwritten again until you re-enable it with RCB bit 7.

Caution

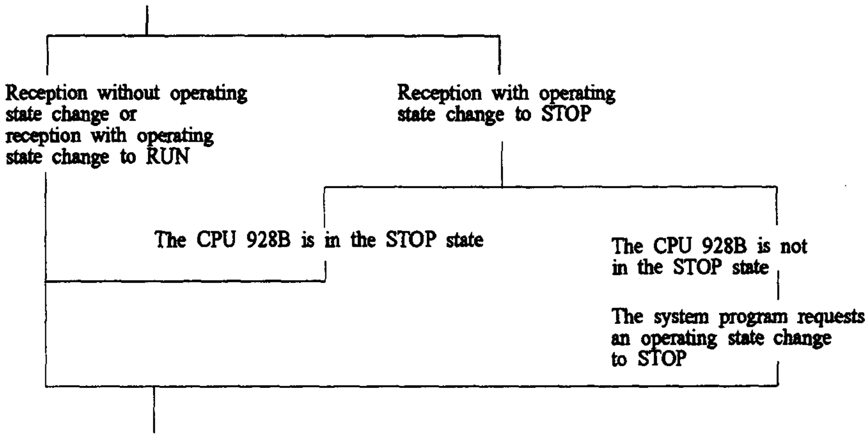
Make sure that your user program processes the received data as quickly as possible and that the receive mailbox is enabled again immediately afterwards by means of RCB bit 7.

The following diagram shows which situations may occur when data reception is in progress in the CPU 928B and the CPU 948.

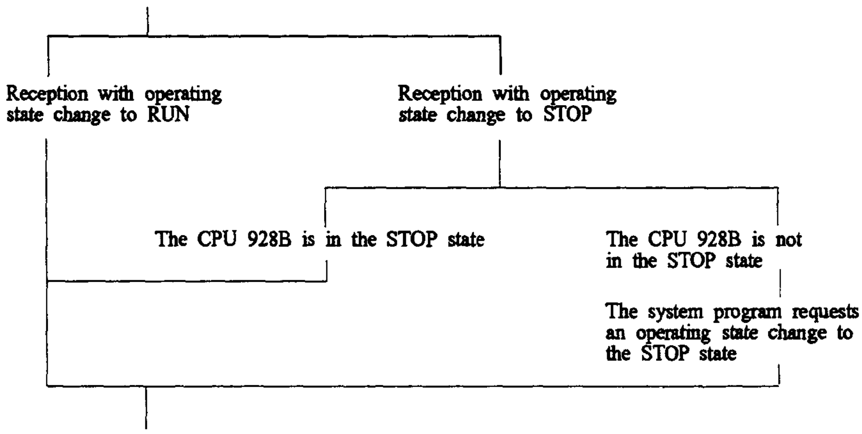
CPU 928B


Reception in progress
The CPU 928B is not in the wait state (no program test)
The CPU 928B is in the wait state (program test)
The system program has
•aborted the receive job with error
•called up OB 35, ACCU1 contains the error code
Reception without operating state change or reception with operating status change to STOP
Reception with operating state change to RUN
The CPU 928B is in the RUN state
The CPU 928B is not in the RUN state
The system program requests an operating state change to RUN
The CPU 928B is ready to start again
The CPU 928B is not ready to start again
The system program has
•set bit 0 in RCB
•updated bits 1 and 2 in RCB
•aborted receive job with error
•called up OB 35, ACCU 1 contains the error code
User data have been received
No user data have been received
Reception without operating state change
Reception with operating state change
Receive mailbox disabled, RCB bit 7 set
Receive mailbox disabled, RCB bit 7 reset
See Section (2)
Reception terminated without error
Reception terminiated with an error
See Section (1)
The system program has
•transferred the received data to receive mailbox from DW1 onwards
•entered the number of received data (specified in bytes) in DL 0 of receive mailbox
•entered node number of source DR 0 of receive mailbox
•reset bit 7 in RCB
•updated bits 1, 2 and 4 in RCB
•reset bit 7 in RCB
The system program has
•set bit 0 in RCB
•updated bits 1 and 2 in RCB
•aborted receive job with error
•called up OB 35, ACCU1 contains the error code
The system program has
•reset bit 0 in RCB
•updated bits 1 and 2 in RCB

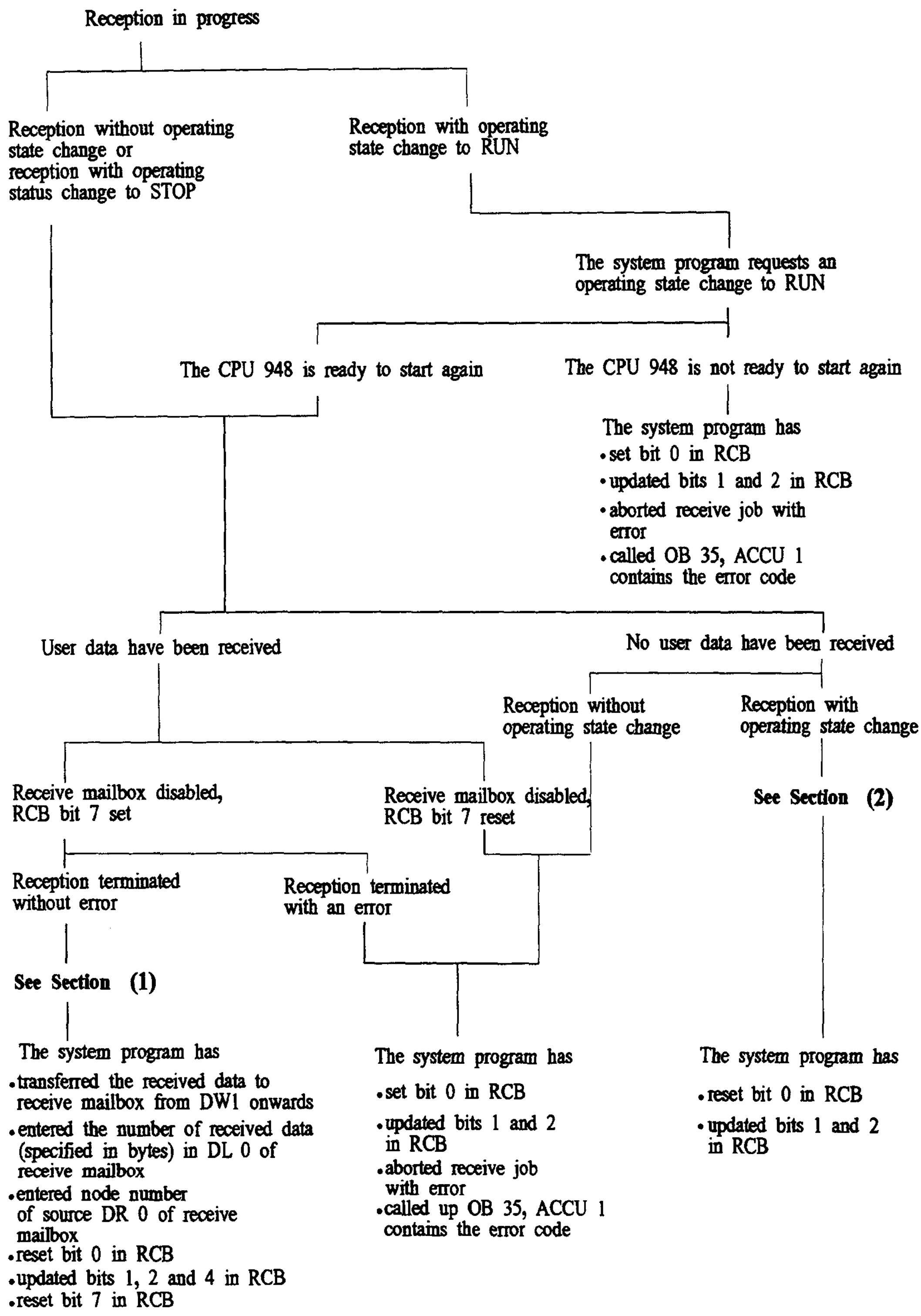
Section (1)



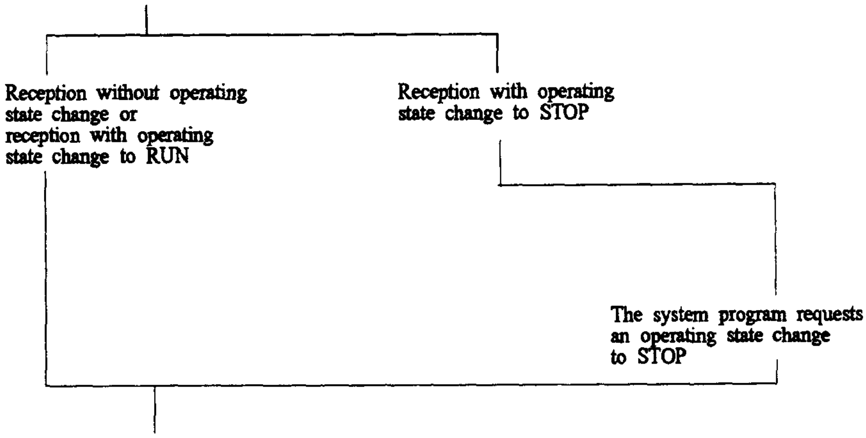
Section (2)



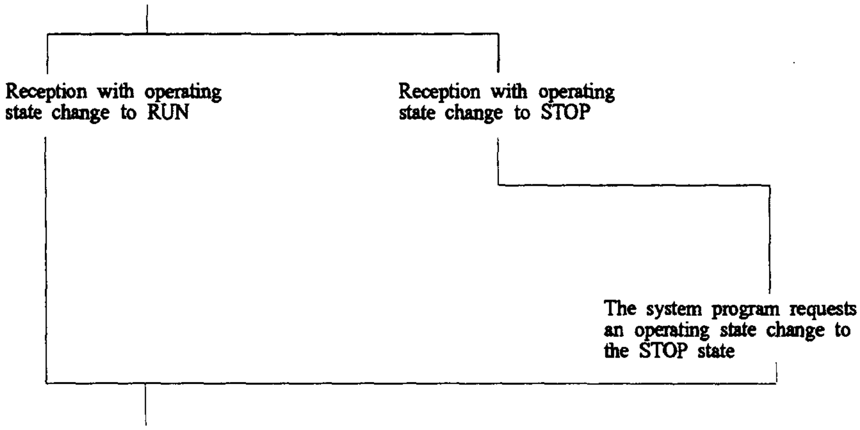
CPU 948


Reception in progress
Reception without operating state change or reception with operating status change to STOP
Reception with operating state change to RUN
The system program requests an operating state change to RUN
The CPU 948 is ready to start again
The CPU 948 is not ready to start again
The system program has
• set bit 0 in RCB
• updated bits 1 and 2 in RCB
• aborted receive job with error
• called OB 35, ACCU 1 contains the error code
User data have been received
No user data have been received
Reception without operating state change
Reception with operating state change
Receive mailbox disabled, RCB bit 7 set
Receive mailbox disabled, RCB bit 7 reset
See Section (2)
Reception terminated without error
Reception terminated with an error
See Section (1)
The system program has
• transferred the received data to receive mailbox from DW1 onwards
• entered the number of received data (specified in bytes) in DL 0 of receive mailbox
• entered node number of source DR 0 of receive mailbox
• reset bit 0 in RCB
• updated bits 1, 2 and 4 in RCB
• reset bit 7 in RCB
The system program has
• set bit 0 in RCB
• updated bits 1 and 2 in RCB
• aborted receive job with error
• called up OB 35, ACCU 1 contains the error code
The system program has
• reset bit 0 in RCB
• updated bits 1 and 2 in RCB

Section (1)



Section (2)



In order to evaluate reception in the RCB, please note the following program sequence in your user program:

Check RCB bit 7

RCB bit 7 is set

No data have been entered in the receive mailbox

The receive mailbox is enabled

RCB bit 7 is reset

No error has occurred during reception

- Bits 1, 2 and 4 in the RCB are updated
- The received data are stored in the receive mailbox from DW1 onwards
- The number of received data (specified in bytes) is stored in DL0
- The source node number is stored in DR0

You can now process the data

Enable the receive mailbox again via RCB bit 7

7.4.3 System Behavior in Special Operating Modes

Stop mode

CPU 928B

The user program is not processed in the stop mode, i.e. you cannot activate a new send job.

If data are received in the stop mode, they are transferred to the receive mailbox provided it is enabled by RCB bit 7. However, the user program cannot enable the receive mailbox again via RCB bit 7 again afterwards. Any further data reception in the stop mode is rejected and acknowledged as an error in the LAN.

CPU 948

In contrast to the CPU 928B, the CPU 948 also supports communication in "soft" stop and in the wait mode (program test).

Wait mode

CPU 928B

The CPU 928B switches to the wait mode when the programmer on-line function "PROGRAM TEST" is active.

The user program is not processed in the wait mode, i.e. you cannot activate a new send job. Any job in processing which has been initiated by the user program will be continued and terminated when the system exits from the wait mode.

If data are received when the system is in wait, the reception is negatively acknowledged in the LAN. A corresponding error message is entered in the interface error area (see Chapter 11, "Error Messages"). If the system switches to the wait mode when data reception is in progress, then the reception is negatively acknowledged in the LAN. A corresponding error message is entered in the error area of the interface (see Chapter 11, "Error Messages").

CPU 948

In contrast to the CPU 928B, the CPU 948 also supports communication in "soft" stop and in the wait mode (program test).

Error Messages

11

Contents of Chapter 11

Error Messages

11

This chapter describes the response to errors during communication on the second interface of the CPU 928B/CPU 948 when using the following:

- RK 512 computer link
- data transmission with the 3964/3964R procedures
- data transmission with the "open driver"
- data transmission with SINEC L1.

This description is followed by an explanation of the errors represented by the error numbers as well as information about how to remedy the error.

PG errors on the second interface are not signalled.

11.1 How the Communications Functions Respond to Errors

Every 100 ms, the system program checks whether communications errors have occurred on the second serial interface. If errors have occurred, the system program calls **OB 35**. You can program the reaction to communications errors in OB 35. Remember that OB 35 is called asynchronously to the user program.

CPU 928B

The defaults in the extended data block DX 0 determine whether or not the CPU changes to the stop mode if OB 35 is not loaded (standard setting: no system stop). If the CPU changes to the stop mode, the error number(s) for the second serial interface are written to ACCU 1.

> **Note**
> Remember that OB 35 is part of the user program and is not processed in the STOP mode. If communications errors occur on the second interface during the STOP mode, the error is indicated via OB 35 during the warm or cold restart.

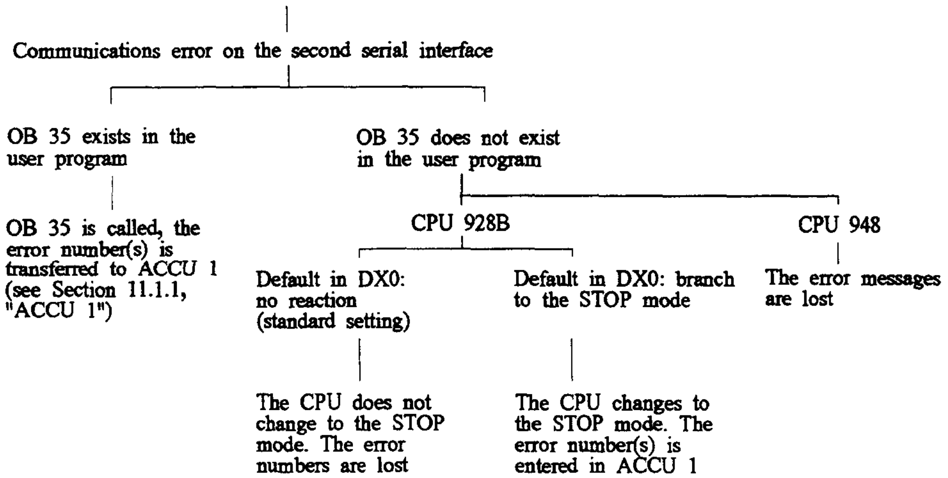
CPU 948

The CPU 948 remains in the RUN mode if OB 35 is not present. The error messages are lost.

> **Note**
> In contrast to the CPU 928B, OB 35 is also processed in "soft" stop in the CPU 948.

If OB 35 is called, ACCU 1 contains information about the errors which can be interpreted directly by your user program.

The following diagram illustrates the response of the system program if communications errors occur on the second serial interface.



11.1.1 ACCU 1

Error numbers for a maximum of three faults/errors can be transferred when OB 35 is called. If there are more than three faults/errors, this is indicated by a special overflow ID.

Structure of the error information in ACCU 1:

ACCU 1:	0	0	0	0	E	O	B	0	Error number 1	Error number 2	Error number 3
	31							24	23 16	15 8	7 0

E =0 no error entered in the error area
=1 error(s) entered in the error area

O =0 no error overflow (max. 3 errors entered)
=1 error overflow (more than 3 errors entered)

B =0 no BREAK on the second serial interface
=1 BREAK on the second serial interface

Each time OB 35 is called by the system program, new error numbers are transferred to ACCU 1. You can evaluate these error numbers and react to them in OB 35.

If there is a BREAK on the second serial interface, OB 35 is only called at the beginning of the BREAK state. A BREAK state on an interface does not cause OB 35 to be called every 100 ms.

Example

In OB 35, the error area of the interface SI 2 is copied into DW 10 and DB 11.

OB 35

```
C    DB10    ;    call data block 10
T    DD10    ;    transfer ACCU 1 to DB 10
```

Result:

DL								DB 10 DR	
0	0	0	0	F	U	B	0	Error number 1	DW 10
Error number 2								Error number 3	DW 11

Caution
Remember that the next time OB 35 is called by the system program, the data double word DD 10 in this example is overwritten by a new error entry.

11.1.2 Special Features with RK 512

If a SEND or FETCH job is terminated with an error, an error number is entered in bits 8 to 11 of the condition codeword (see Section 4.5.5).

If the CPU 928B or the CPU 948 detects an error in the partner's command message, it informs the partner of the error in the reply message (in the fourth byte).

If the partner recognizes an error when processing a CPU 928B/ CPU 948 job, it informs the CPU 928B/CPU 948 of the error in the reply message (in the fourth byte).

If a parameter assignment error is detected in the job mailbox for a SEND or FETCH job, an error number is indicated in the PAFE byte.

In all these cases, an error number is also indicated in ACCU 1.

11.2 Error Lists

The error numbers supplied by the system program are listed according to four criteria, as follows:

- ACCU 1 error numbers, Section 11.2.1
- CC error numbers (only RK 512), Section 11.2.2
- REPMES error numbers (only RK 512), Section 11.2.3
- PAFE error numbers (only RK 512), Section 11.2.4

The error description in the list of ACCU error numbers begins with a shaded line, containing the following abbreviations:

RK 512 computer link	RK
Data transmission with the 3964/3964R procedure	3964/R
Data transmission with the "open driver"	OD
Data transmission with SINEC L1	SINEC L1

Example: RK 3964/R

means that the error description and remedy apply to the RK 512 and the 3964/3964R procedures.

11.2.1

ACCU Error Numbers

Error number ACCU 1	only RK 512 CC	only RK 512 REPMES	Error description	Remedy
01H	01H	0AH	**RK**	
			DW no. in job mailbox for data source or data dest. outside the permitted values.	Correct the parameter assignment of the job mailbox, the permitted values are listed in Section 4.2.4.
			Specified length in the job mailbox for data source or data dest. outside the permitted values.	
			Source/dest. type not permitted or area not permitted.	Partner supplies illegal parameters in the message header. Correct the parameter ass. on the partner.
	-	-	**3964/R OD SINEC L1**	
			Transmit or receive mailbox too long.	Correct the length of the transmit or receive mailbox in DX 2.
02H	02H	14H	**RK**	
			Job mailbox does not exist.	Set up the job mailbox in a DB or DX.
			Data source or destination in the job mailbox does not exist or is not permitted (e.g. the following are not permitted DB 0, DB 1, DB 2, DX 0, DX 1, DX 2).	Correct the parameter assignment for the job mailbox. The permitted values are listed in Section 4.2.4. Set up the data source or data destination in a permitted block.
			Data destination or source of the partner job does not exist.	Correct the parameter assignment for the job on the partner or set up the data destination or the data source of the partner job in a permitted block.
	-	-	**3964/R OD SINEC L1**	
			Transmit or receive mailbox does not exist.	Set up the transmit or receive mailbox in a permitted DB or DX.
03H	03H	14H	**RK**	
			Job mailbox too short or data source or destination for the job mailbox too short.	Correct the block length.
			Data destination or source of the partner job too short.	Correct the block length.
	-	-	**3964/R OD SINEC L1**	
			Transmit or receive mailbox too short.	Correct the block length or correct the parameters for the length of the transmit or receive mailbox in DX 2.
04H			**SINEC L1**	
			Receive mailbox disabled with RCB bit 7	Ensure that your user program evaluates your received data as quickly as possible and then enable the receive mailbox via RCB bit 7 again as quickly as possible.

Error number			Error description	Remedy
ACCU 1	only RK 512			
	CC	REPMES		
05H	-	-	RK Condition codeword does not exist or not permitted.	Correct the assignment of the condition codeword in the job mailbox.
06H	-	-	RK 3964/R OD SINEC L1 Coordination bytes do not exist.	Check the coordination bytes assigned in DX 2 exist.
07H	-	-	RK PAFE byte does not exist in job mailbox or not permitted.	Correct the assignment of the PAFE byte in the job mailbox.
12H	06H	0A	RK 3964/R OD SINEC L1 Error in communication between the operating system proc./standard proc. and communications proc.	Execute a cold restart. If necessary, replace the module.
13H	06H	-	RK Incorrect parameter assignment for the job mailbox. Wrong job type. Illegal job number Type of data source or data destination wrong	Assign correct parameters for the job mailbox: job type "A" for SEND or "E" for FETCH, values from 1 to 255 are permitted as the job number, The data source or data destination are in a data block (DB) or extended data block (DX): 44 ("D") for DB or 58 ("X") for DX.
14H	06H	0A	RK 3964/R OD SINEC L1 Error in communication between the communications processor and operating system proc./standard proc.	Execute a cold restart. If necessary, replace the module.
16H	07H	-	RK 3964/R OD Current job was aborted as the CPU changed to the STOP mode. during the data transfer between the dual-port RAM and user memory (refer to Section 1.4)	Switch the CPU 928B to RUN.
18H	07H	-	RK 3964/R OD SINEC L1 No communication possible	Execute a cold restart. If necessary, replace the module.
20H	07H	-	RK Dynamic parameter set does not exist. Different job types are specified in the job mailbox and in the dynamic parameter set.	Create a dynamic parameter set for the specified job number in the job mailbox or assign a different dynamic parameter set to the job (using the job number in the job mailbox). Specify the same job type: "A" for SEND or "E" for FETCH.

Error number			Error description	Remedy
ACCU 1	only RK 512			
	CC	REPMES		
21H	07H	-	RK The job type and data destination or source are not correct in the dynamic parameter set.	Assign correct parameters to the dynamic parameter set: specify job type "A" for SEND or "E" for FETCH, the data source or destination are in a DB or DX: 44 ("D") for DB or 58 ("X") for DX.
22H	07H	-	RK The DB/DX no. or the DW no. of the data destination or source in the dynamic parameter set are outside the permitted values.	Correct the parameter assignment for the dynamic parameter set, the permitted values are listed in the job tables.
23H	07H	-	RK The bit no. for the coordination flag in the dynamic parameter set is not permitted.	Correct the bit number for the coordination flag, permitted: 0 to 7 or FH (see Section 4.3).
24H	07H	-	RK The CPU number in the dynamic parameter set is not permitted.	Correct the CPU number, permitted: 0, 1, 2, 3, 4 or FH (see Section 4.2.3).
25H	-	-	OD The received message is longer than the receive mailbox or the received message is longer than the fixed message length assigned in the static parameter set when it is received (mode 3). SINEC L1 The received message is longer than the receive mailbox.	Correct the length specification on the partner. Correct the length specification of the partner or check the parameters of the rec. mailbox in DX2
26H	-	-	3964/R OD SINEC L1 The length to be transmitted is longer than the transmit mailbox or the length to be transmitted is 0.	Correct the length specified in the first word of the transmit mailbox or select a longer transmit mailbox in DX 2.
28H	0EH	-	RK Following procedure error: error transmitting the command message.	The procedure error is entered in the error area. Reading out and remedying these errors is the same as described for D8H, D9H, E0H to FFH.
29H	0CH	-	RK Error receiving a reply message: Reception of a reply message without a prior command message. Reception of a follow-on reply to a command message. Reception of a reply message to a follow-on command message.	The error can be signalled following a cold restart on the local PLC or on the partner (normal system restart procedure). During current operation: error message following errors only detected by the partner, or incorrect partner response; the error message possibly did not occur for jobs ≤ 128 bytes.

Error number			Error description	Remedy
ACCU 1	only RK 512			
	CC	REPMES		
2AH	0BH	-	RK Error in reply message from the partner: - 1st byte not 00H or FFH - 2nd byte not 00H - 3rd byte not 00H	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or whether the partner is functioning correctly.
2BH	0BH	-	RK FETCH job: reply message from the partner contains too many data.	
2CH	0BH	-	RK FETCH job: reply message from the partner contains too few data.	
2DH	0BH	-	RK SEND job: reply message from the partner contains data.	
2EH	-	-	RK 3964/R OD At the start of the communication or during the RESET job, an illegal assignment of the static parameters (priority, mode, ...) is detected. The interface could not be initialized. Hardware fault when the interface was initialized	Correct the illegal parameter assignment and execute a cold restart or a reset. Replace the module if necessary
			SINEC L1 At the start of communication an illegal assignment of the SINEC L1 LAN system node numbers is detected. The interface could not be initialized. Hardware fault when the interface was initialized	Correct the illegal parameter assignment and execute a cold restart or a reset. Replace the module if necessary.
2FH	0DH	-	RK FETCH or SEND job: no reply message was received from the partner during the monitoring time The monitoring time depends on the transmission speed (bps): from 1200 bps 5 sec 600 bps 7 sec 300 bps 10 sec 150 bps 15 sec 110 bps 20 sec	Check the communications link and the partner's status or check whether the partner is too slow.

11

Error number ACCU 1	only RK 512 CC	only RK 512 REPMES	Error description	Remedy
30H	09H	32H	**RK** On the partner: DB or DX disabled by coord. flag.	Reset the coordination flag on the partner and repeat the job.
31H	0AH	0AH	**RK** On the partner: hardware error.	Check the hardware setup of the partner and switch it to the RUN mode.
32H	0AH	0CH	**RK** On the partner: memory access error.	Check that the required data area exists on the partner and is sufficiently long or check the dynamic parameter set and the specified length in the job mailbox.
33H	0AH	14H	**RK** On the partner: DB or DX does not exist or too short.	Check that the required data area exists on the partner and is sufficiently long or check the dynamic parameter set and the specified length in the job mailbox.
34H	0BH	16H	**RK** Error in message header recognized by the partner: 3rd byte in the message header is wrong (correct: "A" or "E" or "0")	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line).
35H	0BH	10H	**RK** Error in message header recognized by the partner: 4th byte in the message header is wrong (correct: "D" or "X")	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line).
36H	0BH	34H	**RK** Partner detects incorrect message length.	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line).
37H	0CH	36H	**RK** Partner detects synchron. error: order of the messages incorrect.	This error can be signalled after a cold restart on the local PLC or on the partner (normal restart response of the system). During current operation: error message following errors only detected by the partner, or incorrect response of the partner; the error message possibly does not occur with jobs ≤ 128 bytes.
38H	0CH	2AH	**RK** No cold restart has been performed on the partner. Partner is in STOP mode or wait state.	Execute a cold restart on the partner. Switch the partner to RUN.
39H	0BH	12H	**RK** Partner signals "system command unknown".	Check whether the partner is functioning correctly.

Error number			Error description	Remedy
ACCU 1	only RK 512			
	CC	REPMES		
3AH	0BH	-	RK Reply message received with unknown error code (4th byte).	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or whether the partner is functioning correctly.
40H	-	10H	RK Error in the command message of the partner: 1st byte is not 00H or FFH	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or whether the partner is functioning correctly.
41H	-	16H	RK Error in the command message of the partner: 3rd byte is not "A" or "E" or "0"	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or whether the partner is functioning correctly.
42H	-	16H	RK Error in the follow-on command message of the partner: letter in the 3rd byte is not as in the 1st command message.	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or whether the partner is functioning correctly.
43H	-	10H	RK Error in the command message of the partner: 4th byte is not "D" or "X"	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or whether the partner is functioning correctly.
44H	-	10H	RK Error in the follow-on command message of the partner: letter in the 4th byte is not as in the 1st command message.	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or whether the partner is functioning correctly.
45H	-	14H	RK Error in the command message of the partner: DB/DX no. in 5th byte illegal.	Refer to the job tables for the permitted DB/DX numbers.
47H	-	34H	RK Error in the command message of the partner: DB/DX no. in the 7th and 8th bytes illegal.	Refer to the job tables for the permitted length.
48H	-	0CH	RK Error in the command message of the partner in the 9th and 10th bytes: a coordination flag is not permitted for this data type or the bit number of the coordination flag is too high.	Refer to the job tables to see whether a coordination flag is permitted. Permitted: bit no. 0 to 7 or FH.

Error number			Error description	Remedy
ACCU 1	only RK 512			
	CC	REPMES		
4BH	-	34H	**RK**	
			SEND message is longer than expected, i.e. more data were received than specified in the message header.	Correct the user program of the partner.
		-	**OD**	
			The end characters did not occur within the maximum permitted length! (Modes 2 and 4)	Correct your user program or the user program of the partner.
4CH	-	34H	**RK**	
			SEND messsage is shorter than specified in the message header or < 128 bytes.	Correct the user program of the partner.
4DH	-	34H	**RK**	
			FETCH message of the partner contains user data.	Correct the incorrect partner function.
4EH	-	2AH	**RK**	
			Command message received while the CPU 928B is in the STOP mode. Command message received while the CPU 928B is in the wait state. Active partner job was aborted by the CPU changing to the STOP mode. Active partner job was aborted by the RESET job	Switch the CPU 928B to RUN.
			3964/R OD	
			Command message received while the CPU 928B is in the STOP mode. Command message received while the CPU 928B is in the wait state. Active partner job was aborted by the CPU changing to the STOP mode. Active partner job was aborted by the RESET job. Partner job was aborted after transfer from the communications processor to the operating system processor by "reset interface".	Switch the CPU 928B to RUN.
			SINEC L1	
			Message received while the CPU 928B in wait state	Switch the CPU 928B to RUN.
4FH			**SINEC L1**	
			A mode state change from STOP to RUN has been requested, but the CPU cannot be restarted.	Switch the operating mode switch to the RUN position or replace the module if necessary.

Error number			Error description	Remedy
ACCU 1	only RK 512			
	CC	REPMES		
50H	-	-	RK After procedure error: error transmitting the reply message.	The procedure error is entered in the error area. Reading out and remedying this error is the same as described for E0H to FFH.
51H	-	36H	RK Command messages of the partner received in the wrong order: Two (follow-on) command messages received, without a (follow-on) reply message being sent between them. Command message received although follow-on command message was expected. Follow-on command message received, although command message was expected.	The error can be signalled following a cold restart on the local PLC or on the partner (normal restart response of the system). During current operation: error message following errors only detected by the partner or incorrect response of the partner; the error message possibly does not occur with jobs ≤ 128 bytes.
52H	-	32H	RK DB/DX disabled by coordination flag.	In your own user program: reset the coordination flag, once all the transmitted data have been processed. On the partner: repeat the job.
5EH	-	-	RK The message is too short (with follow-on or reply messages < 4 bytes, with command messages < 10 bytes), the message is ignored, the currently active job is, however, not aborted.	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or whether the partner is functioning correctly.
5FH	-	34H	RK The command message from the partner is too long. The job is aborted.	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or whether the partner is functioning correctly.
80H	-	-	SINEC L1 Illegal bit combination in SCB	Check the bit combination in the SCB when the job is initiated (no interrupt communication allowed in point-to-point master).
81H	-	-	SINEC L1 Illegal destination address in DR0 of transmit mailbox	Correct the target address in DR0 of the transmit mailbox.

11

Error number ACCU 1	only RK 512 CC	only RK 512 REPMES	Error description	Remedy
82H	-	-	SINEC L1 LAN monitoring time lapsed	Check whether there is interference in the LAN (with an interface test unit switched into the transmission line), whether one of the other nodes is functioning incorrectly or whether preceded by incorrect functioning of your node (previous error number in ACCU 1).
83H	-	-	SINEC L1 SEND job aborted in BUS OFF state	Put SINEC L1 LAN into RUN state and restart your SEND job.
85H	-	-	SINEC L1 Point-to-point master: No NUM ack. received from slave Slave: With internode comms: no NUM ack. received from passive slave	Check the parameter assignment of the slave. It should have the node number 01H. Check the parameter assignment of the passive slave.
86H	-	-	SINEC L1 Point-to-point master: A character other than the NUM acknowledgement has been received. Slave: With internode comms: A character other than the NUM acknowledgement has been received.	Check whether the partner is functioning correctly or whether there is interference on the transmission line (with an interference test unit switched into the transmission line).
87H	-	-	SINEC L1 Point-to-point master: the NUM acknowledgement has been received with a low-level error (parity, framing...). Slave: With internode comms: the NUM ack. has been received with a low-level error (parity, framing...).	Check whether there is interference on the transmission line (with an interference test unit switched into the transmission line).
88H	-	-	SINEC L1 Characters have been illegally received while characters were transmitted.	Check whether there is interference on the transmission line (with an interference test unit switched into the transmission line).
8EH	-	-	SINEC L1 The coordination flag has aborted the protocol.	Check whether the partner is functioning correctly or whether there is interference on the transmission line (with an interference test unit switched into the transmission line).

Error number			Error description	Remedy
ACCU 1	only RK 512			
	CC	REPMES		
90H	-	-	SINEC L1 NUM+ was received with low-level error (parity, framing...) while the master was establishing a connection.	Check whether there is interference on the transmission line (with an interference test unit switched into the transmission line).
91H	-	-	SINEC L1 NUM+ was received with an illegal call identifier while the master was establishing a connection.	Check whether the master is functioning correctly or whether there is interference on the transmission line (with an interface test unit switched into the transmission line)
92H	-	-	SINEC L1 NUM+ was received with the address of the master while the master was establishing a connection.	Check whether the master is functioning correctly or whether there is interference on the transmission line (with an interface test unit switched into the transmission line)
93H	-	-	SINEC L1 Call identifier set in PG function during master addressing	Check whether the partner is functioning correctly or whether there is interference on the transmission line (with an interface test unit switched into the transmission line)
94H	-	-	SINEC L1 Incorrect master pole end identifier received	Check whether the partner is functioning correctly or whether there is interference on the transmission line (with an interface test unit switched into the transmission line)
95H	-	-	SINEC L1 A low-level error (parity, framing) was recognized while receiving a character.	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line).
96H	-	-	SINEC L1 The length of the user data received in the message header is longer than 64	Check whether the partner has transmitted the wrong length or whether there is interference on the transmission line (with an interface test unit switched into the transmission line).
97H	-	-	SINEC L1 The calculated BCC1 (message header or acknowledgement) does not correspond to the received BCC1.	Check whether the transmitted BCC1 corresponds to the specifications or whether there is interference on the transmission line (with an interface test unit switched into the transmission line).
98H	-	-	SINEC L1 The calculated BCC2 (user data) does not correspond to the received BCC1.	Check whether the transmitted BCC2 corresponds to the specifications or whether there is interference on the transmission line (with an interface test unit switched into the transmission line).

11

Error number			Error description	Remedy
ACCU 1	only RK 512			
	CC	REPMES		
99H	-	-	SINEC L1 Point-to-point master: The slave has sent a negative implicit acknowledgement because it could not receive the data sent to it. Slave: The master (normal communication) or the passive slave (internode communication) has sent a negative implicit acknowledgement because could not receive the data sent to it.	Check why the partner cannot receive the data (e.g. because no receive mailbox exists) or whether there is interference on the transmission line (with an interface test unit switched into the transmission line)
9AH	-	-	SINEC L1 Point-to-point master: The acknowledgement byte sent by the slave does not correspond to the specification: Bit 7: always 1 Bit 6: 0 = pos./1 = neg. acknowledgement Bit 5: always 0 Bits 0-4: no. of source of ack. Slave: The acknowledgement byte sent by the master (normal comm.) or the passive slave (internode comm.) does not correspond to the specification: Bit 7: always 1 Bit 6: 0 = pos./1 = neg. acknowledgement Bit 5: always 0 Bits 0-4: no. of source of ack.	Check why the partner has sent an incorrect acknowledgment or whether there is interference on the transmission line (with an interface test unit switched into the transmission line).
9BH	-	-	SINEC L1 Point-to-point master: The slave has not sent the number of the master (0) as the target number in the message!	Check whether the partner is functioning correctly or whether there is interference on the transmission line (with an interface test unit switched into the transmission line).
9EH	-	-	SINEC L1 Slave: With internode communication: An impermissibly large COUNTER BREAK was recognized while the active slave was establishing a conn.	Check whether the partner is functioning correctly or whether there is interference on the transmission line (with an interface test unit switched into the transmission line).
9FH	-	-	SINEC L1 Reception line BREAK although data is expected	Check whether the partner is functioning correctly or whether there is interference on the transmission line (with an interface test unit switched into the transmission line).

Error number			Error description	Remedy
ACCU 1	only RK 512			
	CC	REPMES		
AAH	-	-	RK 3964/R OD SINEC L1	
			Internal error on the communications processsor.	Execute a cold restart. If necessary, replace the module.
C0H	-	-	RK 3964/R OD SINEC L1	
			Checksum error in the EPROM of the communications system (incorrect or defective).	Replace the module.
C1H	-	-	RK 3964/R OD SINEC L1	
			The required communication cannot be loaded.	Check the parameter assignment in DX 2. Execute a cold restart/overall reset. If necessary, replace the module.
C2H	-	-	RK 3964/R OD SINEC L1	
			RAM error in the memory area of the communications processor.	Replace the module.
C6H	-	-	RK 3964/R OD SINEC L1	
			The required communication does not exist or the EPROM is empty or missing.	Correct the parameter assignment in DX 2 or replace the module.
C8H	-	-	RK 3964/R OD SINEC L1	
			The interface module is not suitable or not plugged in for the assigned communication.	Check whether the (correct) interface submodule is plugged in.
C9H	-	-	RK 3964/R OD SINEC L1	
			A job for a communications process which is not loaded has been triggered.	Assign parameters in DX 2 for the required communication and execute a cold restart.
D8H	0EH	-	RK 3964/R	
			No connection establishment possible, the number of connection attempts has been exceeded.	Check the interface cable or check the transmission parameters.
D9H	0EH	-	RK 3964/R	
			The data could not be transmitted, the permitted number of repetitions has been exceeded.	Check the interface cable or check the transmission parameters.
E0H	-	-	RK 3964/R	
			The first attempt to transmit the data failed, a repetition is started.	The repetition means interference on the transmission line or that the partner is functioning incorrectly. If, despite repetitions, the data are not transmitted, an error number is entered that describes the error in greater detail.

11

Error number ACCU 1	only RK 512 CC	only RK 512 REPMES	Error description	Remedy
E1H	0EH	-	RK 3964/R During the connection establishment (STX sent) the partner does not reply with DLE or there is an initialization conflict or the link partner refuses to establish a connection with NAK.	Check whether the partner is also indicating an error or whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or check the priority set on the communications partner. The partners must have different priority.
E2H	0EH	-	RK 3964/R During the connection establishment (STX sent) the partner does not reply within the acknowledgement delay time.	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or check the priority set on the communications partner. The partners must have different priority.
E3H	0EH	-	RK 3964/R The partner aborts the data exchange during a current transmission.	Check whether the partner is also indicating an error or whether there is interference on the transmission line (with an interface test unit switched into the transmission line).
E4H	0EH	-	RK 3964/R During connection termination (DLE ETX sent) the partner did not reply with DLE, but with NAK or any other character other than DLE.	Check whether the partner is also indicating an error (for example due to heavy interference) or whether there is interference on the transmission line (with an interface test unit switched into the transmission line).
E5H	0EH	-	RK 3964/R During connection termination (DLE EXT sent), no reply was received from the partner during the acknowledgement delay time.	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or whether the partner is functioning correctly or whether the partner is too slow.
F0H	-	-	RK 3964/R 1st attempted repetition is active or expected, since the CPU 928B detected an error while receiving a message and requested a repetition (with NAK).	The repetition means interference on the transmission line or that the partner is functioning incorrectly. If, despite repetitions, the data are not transmitted, an error number is entered that describes the error in greater detail.
F1H	-	-	RK 3964/R Characters were received in the idle state, not STX or NAK or during connection establishment the partner sent further characters following STX without waiting for the reply DLE.	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or whether the partner is functioning correctly.

Error number			Error description	Remedy
ACCU 1	only RK 512			
	CC	REPMES		
F2H	-	-	RK 3964/R Other characters received after DLE, after DLE, only DLE (DLE doubling) or ETX (connection termination) can follow.	Check whether the partner has doubled DLE in the message header and in the data string and whether the connection termination is indicated by DLE ETX. Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line).
F3H	-	-	RK 3964/R OD The character delay time between two characters was exceeded.	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or whether the partner is functioning correctly.
F4H	-	-	RK 3964/R Missing block check character (BCC) or Error in the block check character: the internally generated value of the BCC does not match the value sent by the partner at the end of the connection.	Check whether the connection is impaired, in which case the error number FEH is also indicated or whether the partner is functioning correctly.
F5H	-	-	RK 3964/R Wait time for message repetition elapsed. No message repetition was received from the partner within a wait time of 4s.	Check whether the partner is functioning correctly.
F6H	-	-	3964/R No free receive buffer available, not even after waiting for 400 ms.	Release the receive mailbox with the RCB.
			OD No free receive buffer available, not even after waiting for 400 ms.	Release the receive mailbox using RCB. Use the flow control.
F8H	-	-	RK 3964/R End identifier DLE ETX not recognized or the data frame is too long.	The length of useful data frames is a maximum of 128 bytes for the RK 512 computer link, and maximum 4096 bytes for transmission with 3964/3964R. Check the length given for the receive mailbox in DX 2.

Error number			Error description	Remedy
ACCU 1	only RK 512			
	CC	REPMES		
FEH	-	-	RK 3964/R OD	
			Transmission error (parity error, stop bit error, overflow error)	Interference on the transmission line causes message repetitions and reduces the data throughput. The risk of an undetected error is increased. Change your system structure. Check whether the transmission speed, parity and number of stop bits is set the same on both devices.
FFH	0FH	-	RK 3964/R OD	
			BREAK the connecting cable to the partner is interrupted.	Re-establish the connection to the partner, switch the partners to RUN.
	-	-	SINEC L1	
			BREAK, the connection to the LAN master is interrupted.	Establish the connection from the interface to the LAN master.

11.2.2 CC Error Numbers

An error number is only entered in the condition codeword (CC) when the error occurs during the processing of a SEND or FETCH job.

Error number			Error description	Remedy
CC	ACCU 1	REPMES		
01H	01H	0AH	DW no. in job mailbox for data source or data dest. outside the permitted values. Specified length in the job mailbox for data source or data dest. outside the permitted values.	Correct the parameter assignment of the job mailbox, the permitted values are listed in Section 4.2.4.
02H	02H	14H	Data source or destination in the job mailbox does not exist or is not permitted (e.g. the following are not permitted DB 0, DB 1, DB 2, DX 0. DX 1, DX 2) DB/DX no. for data source or dest. in the job mailbox is outside the permitted values.	Set up the data source or data destination in a permitted block or correct the parameter assignment of the job mailbox, the permitted values are listed in Section 4.2.4.
03H	03H	14H	Data source or data destination in the job mailbox too short.	Correct the block length.
06H	12H	0AH	Error in communication between the operating system processor and the communications processor.	Execute a cold restart, if necessary replace the module.
	13H	-	Incorrect parameter assignment for the job mailbox: wrong job type illegal job number type of data source or data destination wrong.	Assign correct parameters for the job mailbox: job type "A" for SEND or "E" for FETCH, values from 1 to 255 are permitted as the job no., the data source or data destination are in a data block (DB) or extended data block (DX): 44 ("D") for DB or 58 ("X") for DX.
	14H	0AH	Error in communication between the communications processor and operating system processor.	Execute a cold restart. If necessary, replace the module.
07H	16H	-	Current job was aborted as the CPU changed to the STOP mode during the data transfer between the dual-port RAM and the user memory (refer to Section 1.4)	Switch the CPU 928B to RUN.
	18H	-	No communication possible.	Execute a cold restart. If necessary, replace the module.
	20H	-	Dynamic parameter set does not exist Different job types are specified in the job mailbox and in the dynamic parameter set.	Create a dynamic parameter set for the specified job number in the mailbox or assign a different dynamic parameter set to the job (using the job number in the job mailbox). Specify the same job type: "A" for SEND or "E" for FETCH.

Error number			Error description	Remedy
CC	ACCU 1	REPMES		
07H	22H	-	The DB/DX no. or the DW no. of the data destination or source in the dynamic parameter set are outside the permitted values.	Correct the parameter assignment for the dynamic parameter set, the permitted values are listed in the job tables.
	23H	-	The bit no. for the coordination flag in the dynamic parameter set is not permitted.	Correct the bit number for the coordination flag, permitted: 0 to 7 or FH (see Section 4.3).
	24H	-	The CPU number in the dynamic parameter set is not permitted.	Correct the CPU number, permitted: 0, 1, 2, 3, 4 or FH (see Section 4.2.3).
09H	30H	32H	On the partner: DB or DX disabled by coordination flag.	Reset the coordination flag on the partner and repeat the job.
0AH	31H	0AH	On the partner: hardware error.	Check the installed hardware on the partner and switch it to the RUN mode.
	32H	0CH	On the partner: memory access error.	Check whether or not the required data area exists on the partner and is sufficiently long or check the dynamic parameter set and the length specified in the job mailbox.
	33H	14H	On the partner: DB or DX does not exist or is too short.	Check whether or not the required data area exists on the partner and is sufficiently long or check the dynamic parameter set and the length specified in the job mailbox.
0BH	2AH	-	Error in reply message from the partner: - 1st byte not 00H or FFH - 2nd byte not 00H - 3rd byte not 00H	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or whether the partner is functioning correctly.
	2BH	-	FETCH job: reply message from the partner contains too many data.	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or whether the partner is functioning correctly.
	2CH	-	FETCH job: reply message from the partner contains too few data.	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or whether the partner is functioning correctly.
	2DH	-	SEND job: reply message from the partner contains data.	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or whether the partner is functioning correctly.

Error number			Error description	Remedy
CC	ACCU 1	REPMES		
0BH	34H	16H	Error in message header recognized by the partner: 3rd byte in the message header is wrong (correct: "A" or "E" or "0")	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or whether the partner is functioning correctly.
	35H	10H	Error in message header recognized by the partner: 4th byte in the message header is wrong (correct: "D" or "X")	
	36H	34H	Partner detects incorrect message length.	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or whether the partner is functioning correctly.
	39H	12H	Partner signals "system command not recognized".	Correct the incorrect response of the partner.
	3AH	-	Reply message with unknown error code (4th byte) received	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or whether the partner is functioning correctly.
0CH	29H	-	Error receiving a reply message: reception of a reply message without a prior command message reception of a follow-on reply to a command message reception of a reply message to a follow-on command message	The error can be signalled following a cold restart on the local PLC or on the partner (normal system restart procedure). During current operation: error message following errors only detected by the partner, or incorrect partner response; the error message possibly does not occur for jobs ≤ 128 bytes.
	37H	36H	Partner detects synchron. error: order of the message is incorrect.	The error can be signalled following a cold restart on the local PLC or on the partner (normal system restart procedure). During current operation: error message following errors only detected by the partner, or incorrect partner response; the error message possibly does not occur for jobs ≤ 128 bytes.
	38H	2AH	No cold restart has been performed on the partner.	Execute a cold restart on the partner.
0DH	2FH	-	FETCH or SEND job: no reply message was received from the partner during the monitoring time The monitoring time depends on the transmission speed (bps): from 1200 bps 5 sec 600 bps 7 sec 300 bps 10 sec 150 bps 15 sec 110 bps 20 sec	Check the communications link and the partner's status or check whether the partner is too slow.

Error number			Error description	Remedy
CC	ACCU 1	REPMES		
0EH	28H		Following procedure error: error transmitting the command message.	The procedure error is entered in the error area. Reading out and remedying these errors is the same as described in Section 11.2.1 for E0H to FFH.
	D8H		No connection establishment possible, the number of connection attempts has been exceeded.	Check the interface cable or check the transmission parameters.
	D9H		The data could not be transmitted, the permitted number of repetitions has been exceeded.	Check the interface cable or check the transmission parameters.
0EH	E1H	-	During the connection establishment (STX sent) the partner does not reply with DLE or there is an initialization conflict.	Check whether there is interference on the transmission line (with an interference test unit switched into the transmission line) or check the priority set on the communications partner. The partners must have different priority.
	E2H	-	During the connection establishment (STX sent) the partner does not reply within the acknowledgement delay time.	Check whether there is interference on the tranmission line (with an interface test unit switched into the transmission line) or check the priority set on the communications partner. The partners must have different priority.
	E3H	-	The partner aborts the data exchange during a current transmission.	Check whether the partner is also indicating an error or whether there is interference on the transmission line (with an interface test unit switched into the transmission line).
	E4H	-	During connection termination (DLE ETX sent) the partner did not reply with DLE, but with NAK or any other character other than DLE.	Check whether the partner is also indicating an error (for example, due to heavy interference) or whether there is interference on the transmission line (with an interface test unit switched into the transmission line).
	E5H	-	During connection termination (DLE ETX sent), no reply was received from the partner during the acknowledgement delay time.	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or whether the partner is functioning correctly or whether the partner is too slow.
0FH	FFH	-	BREAK the connecting cable to the partner is interrupted.	Re-establish the connection to the partner, switch the partners to RUN.

11.2.3 REPMES Error Numbers

If an error occurs involving a command message from the partner, an error number is signalled to the partner in the 4th byte of the reply message.

Error number			Error description	Remedy
REPMES	ACCU 1	CC		
0AH	01H	01H	Source/dest. type not permitted or area not permitted.	Partner supplies illegal parameters in the message header, correct the parameter assignment on the partner.
	12H	06H	Error in communication between the operating system processor and communications processor.	Execute a cold restart. If necessary, replace the module.
	14H	06H	Error in communication between the communications processor and operating system processor.	Execute a cold restart. If necessary, replace the module.
0CH	48H	-	Error in command message of the partner in the 9th and 10th bytes: a coordination flag is not permitted with this data type or the bit number of the coordination flag is too high.	Refer to the job tables for permitted use of a coordination flag. Permitted bit number 0 to 7 or FH.
	49H	-	Error in the command message of the partner: CPU no. in 10th byte not permitted.	Permitted: CPU no. 0 to 4 or FH.
10H	40H	-	Error in command message from the partner: 1st byte ist not 00H or FFH	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or whether the partner is functioning correctly.
	43H	-	Error in the command message of the partner: 4th byte is not "D" or "X"	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or whether the partner is functioning correctly.
	44H	-	Error in the follow-on command message of the partner: letter in the 4th byte is not as in the 1st command message.	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or whether the partner is functioning correctly.
14H	02H	02H	Data destination or source of the partner job does not exist.	Set up the data destination or source of the partner job in a permitted block.
	03H	03H	Data source or destination too short.	Correct the block length.
	45H	-	Error in command message of the partner: DB/DX no. in the 5th byte illegal.	Refer to the job tables for the permitted DB/DX numbers.
16H	41H	-	Error in the command message of the partner: 3rd byte is not "A" or "E" or "0"	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or whether the partner is functioning correctly.

Error number			Error description	Remedy
REPMES	ACCU 1	CC		
16H	42H	-	Error in the follow-on command message of the partner: letter in the 3rd byte is not as in the 1st command message.	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or whether the partner is functioning correctly.
2AH	4EH	-	Command message received while the CPU 928B is in the STOP mode. Command message received while the CPU 928B is in the wait state. Active partner job was aborted by the CPU changing to the STOP mode.	Switch the CPU 928B to RUN.
32H	52H	-	DB/DX disabled by coordination flag.	In your own user program: reset the coordination flag, once all the transmitted data have been processed. On the partner: repeat the job.
34H	47H	-	Error in command message of the partner: length in 7th and 8th byte illegal.	Refer to the job tables for the permitted length.
	4BH	-	SEND message is longer than expected, i.e. more data received than indicated in message header.	Correct the user program on the partner.
	4CH	-	SEND message is shorter than indicated in the message header or < 128 bytes	Correct the user program on the partner.
	4DH	-	FETCH message from the partner contains user data.	Correct the user program on the partner.
	5FH	-	The command message of the partner is too long. The job is aborted.	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or whether the partner is functioning correctly.
36H	51H	-	Command messages of the partner received in the wrong order: two (follow-on) command messages received without a (follow-on) reply message being sent between them command message received, although follow-on command message was expected follow-on command received, although command message was expected	The error can be signalled following a cold restart on the local PLC or on the partner (normal restart response of the system). During current operation: error message following errors only detected by the partner or incorrect response of the partner; the error message possibly does not occur with jobs ≤ 128 bytes.

Errors recognized by the partner involving a command message are signalled to the CPU 928B in the reply message with an error number in the 4th byte.

Error number			Error description	Remedy
REPMES	ACCU 1	CC		
0AH	31H	0AH	On the partner: hardware error	Check the check hardware set up of the partner and switch it to the RUN mode.
0CH	32H	0AH	On the partner: memory access error.	Check that the required data area exists on the partner and is sufficiently long or check the dynamic parameter set and the specified length in the mailbox.
10H	35H	0BH	Error in message header recognized by the partner: 4th byte in the message is wrong (correct: "D" or "X")	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or whether the partner is functioning correctly.
12H	39H	0BH	Partner signals "system command unknown".	Check whether the partner is functioning correctly.
14H	33H	0AH	On the partner: DB or DX does not exist or too short.	Check that the required data area exists on the partner and is sufficiently long or check the dynamic parameter set and the specified length in the mailbox.
16H	34H	0BH	Error in message header recognized by the partner: 3rd byte in the message header is wrong (correct: "A" or "E" or "0")	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or whether the partner is functioning correctly.
2AH	38H	0CH	No cold restart has been performed on the partner.	Execute a cold restart on the partner.
32H	30H	09H	On the partner: DB or DX disabled by coordination flag.	Reset the coordination flag on the partner and repeat the job.
34H	36H	0BH	Partner detects wrong message length.	Check whether there is interference on the transmission line (with an interface test unit switched into the transmission line) or whether the partner is functioning correctly.
36H	37H	0CH	Partner detects synchron. error: order of the messages incorrect.	This error can be signalled after a cold restart on the local PLC or on the partner (normal restart response of the system). During current operation: error message following errors only detected by the partner, or incorrect response of the partner; the error message possibly does not occur with jobs ≤ 128 bytes.

11.2.4 PAFE Error Messages

PAFE byte	Description	Remedy
00H	No error	
11H	In the job mailbox: DW no. for data source/destination or length specification outside the limits	Correct the parameter assignment for the job mailbox (see Section 4.2.4).
21H	In the job mailbox: DB or DX no. outside the limits or data source/destination does not exist	Correct the parameter assignment for the job mailbox (see Section 4.2.4).
31H	In the job mailbox: source/destination block too short	Check the length specified in the job mailbox.
41H	Reserved	
51H	In the job mailbox: wrong condition codeword	Check the specifications for the CC in the job mailbox.
61H 71H 81H 91H A1H	Reserved Reserved Reserved Reserved Reserved	
B1H	In the job mailbox: job (SEND/FETCH) different from the dynamic parameter set, job number is illegal or job type unknown	Correct the parameter assignment for the job mailbox (see Section 4.2.4).
C1H D1H E1H F1H	Reserved Reserved Reserved Reserved	

Appendix

12

The Appendix contains

- Ordering data of the products mentioned in the System Manual
- Further Reading
- Index

12

Ordering Data

This section lists the order numbers of the products mentioned or described in the manual.

Interface submodule

for PG/OP	6ES5 752-0AA53
for 20 mA current loop (TTY)	6ES5 752-0AA12
for V.24/V.28	6ES5 752-0AA22
for RS 422-A/485	6ES5 752-0AA43
for SINEC L1	6ES5 752-0AA62
Cover for interface submodule receptacle	C79458-L968-C20
BT 777 bus terminal	6ES5 777-xxx00

Standard connecting cables

(xxx = SIMATIC length key)

CPU 928B/CPU 948 - PG 7xx	6ES5 734-2xxx0
CPU 928B/CPU 948 - CP 544/525/524 (TTY)	6ES5 726-1xxx0
CPU 928B/CPU 948 - CP 544/525/524 (RS 422-A/485)	6ES5 725-7xxx0
CPU 928B/CPU 948 - CP 544/525/524 (V.24)	6ES5 726-8xxx0
CPU 928B/CPU 948 - PT88/PT89 (TTY/V.24)	6ES5 726-5xxx0
CPU 928B/CPU 948 - DR 210/DR 211, DR 230/DR231 (TTY/V.24)	6ES5 726-5xxx0

1.00 m	BB0	63.00 m	CG3
1.60 m	BB6	80.00 m	CJ0
2.00 m	BC0	100.00 m	DB0
2.50 m	BC5	120.00 m	DB2
3.20 m	BD2	160.00 m	DB6
5.00 m	BF0	200.00 m	DC0
8.00 m	BJ0	250.00 m	DC5
10.00 m	CB0	320.00 m	DD2
12.00 m	CB2	400.00 m	DE0
16.00 m	CB6	500.00 m	DF0
20.00 m	CC0	630.00 m	DG3
25.00 m	CC5	800.00 m	DJ0
32.00 m	CD2	1000.00 m	EB0
40.00 m	CE0		
50.00 m	CF5		

Further Reading

/1/ Hans Berger:
Automating with the SIMATIC S5-135U

Siemens AG - Order no. A19100-L531-F505-X-7600

/2/ ST 54.1 catalog S5-135U, S5-155U and S5-155H
Programmable Controllers

/3/ Guidelines for Handling
Electrostatically Sensitive Components

/4/ ST 50 catalog Programmable Controllers

/5/ S5-135U/155U Programmable Controller
CPU 928B Programming Guide
Order No. 6ES5 998-2PR21

/6/ S5-135U Programmable Controller
CPU 948 Programming Guide
Order No. 6ES5 998-3PR21

/7/ S5-135U/155U Programmable Controller
System Manual
Order No. 6ES5 998-0SH21

/8/ "SINEC L1 Bus System" Manual
Order No. 6ES5 998-7LA21

Index

Guidelines for Handling
Electrostatically Sensitive Devices (ESD)

1 What is ESD?

VSLI chips (MOS technology) are used in practically all SIMATIC and TELEPERM modules. These VLSI components are, by their nature, very sensitive to overvoltage and thus to electrostatic discharge:

They are therefore defined as
Electrostatically Sensitive Devices: "ESD"

"ESD" is the abbreviation used internationally.

The following warning label on the cabinets, subracks and packing indicates that electrostatically sensitive components have been used and that the modules concerned are susceptible to touch:

ESDs can be destroyed by voltage and energy levels which are far below the level perceptible to human beings. Such voltages already occur when a component or a module is touched by a person who has not been electrostatically discharged. Components which have been subjected to such overvoltages cannot, in most cases, be immediately detected as faulty; the fault occurs only after a long period in operation.

An electrostatic discharge
- of 3500 V can be felt
- of 4500 V can be heard
- must take place at a minimum of 5000 V to be seen.

But just a fraction of this voltage can already damage or destroy an electric component.

The typical data of a component can suffer due to damage, overstressing or weakening caused by electrostatic discharge; this can result in temporary fault behavior, e.g. in the case of

- temperature variations,
- mechanical shodks,
- vibrations,
- change of load.

Only the consequent use of protective equipment and careful observation of the precautions for handling such components can effectively prevent functional disturbances and failures of ESD modules.

2 When is a Static Charge Formed?

One can never be sure that the human body or the material and tools which one is using are not electrostatically charged.

Small charges up to 100 V are very common; these can, however, very quickly rise up to 35 000 V!

Examples of static charge:

– Walking on a carpet	up to	35 000 V
– Walking on a PVC flooring	up to	12 000 V
– Sitting on a cushioned chair	up to	18 000 V
– Plastic desoldering unit	up to	8 000 V
– Books, etc. with a plastic binding	up to	8 000 V
– Plastic bag	up to	5 000 V
– Plastic coffee cup	up to	5 000 V

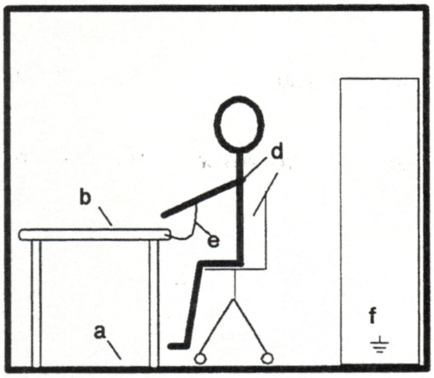
3 Important Protective Measures against Static Discharge

- Most plastic materials are highly susceptible to static charge and must therefore be kept as far away as possible from ESDs!
- Personnel who handle ESDs, the work table and the packing must all be carefully grounded!

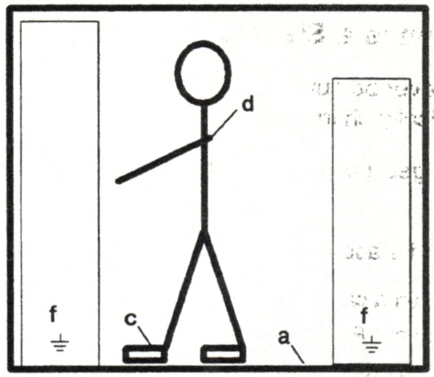
4 Handling of ESD Modules

- One basic rule to be observed is that electronic modules should be touched by hand only if this is necessary for any work to be done on them. Do not touch the component pins or the conductors.
- Touch components only if
  – the person is grounded at all times by means of a wrist strap
  or
  – the person is wearing special anti-static shoes or shoes with a grounding strip.
- Before touching an electronic module, the person concerned must ensure that (s)he is not carrying any static charge. The simplest way is to touch a conductive, grounded item of equipment (e.g. a blank metallic cabinet part, water pipe, etc.) before touching the module.
- Modules should not be brought into contact with insulating materials or materials which take up a static charge, e.g. plastic foil, insulating table tops, synthetic clothing, etc..
- Modules should only be placed on conductive surfaces (table with anti-static table top, conductive foam material, anti-static plastic bag, anti-static transport container).
- Modules should not be placed in the vicinity of visual display units, monitors or TV sets (minimum distance from screen > 10 cm).

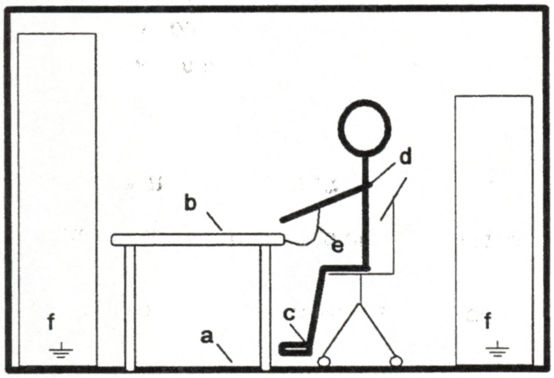
The diagram on the next page shows the required protective measures against electrostatic discharge.



Sitting position



Standing position



a Conductive flooring
b Anti-static table
c Anti-static shoes
d Anti-static coat
e Grounding wrist strap
f Grounding connection of the cabinets

Standing/sitting position

5 Measurements and Modifications to ESD Modules

- Measurements on modules may only be carried out under the following conditions:
  - the measuring equipment is grounded (e.g. via the PE conductor of the power supply system) or
  - when electrically isolated measuring equipment is used, the probe must be discharged (e.g. by touching the metallic casing of the equipment) before beginning measurements.
- Only grounded soldering irons may be used.

6 Shipping of ESD Modules

Anti-static packing material must always be used for modules and components, e.g. metalized plastic boxes, metal boxes, etc. for storing and dispatch of modules and components.

If the container itself is not conductive, the modules must be wrapped in a conductive material such as conductive foam, anti-static plastic bag, aluminum foil or paper. Normal plastic bags or foils should not be used under any circumstances.

For modules with built-in batteries ensure that the conductive packing does not touch or short-circuit the battery connections; if necessary cover the connections with insulating tape or material.

Remarks Form

Your comments and recommendations will help us to improve the quality and usefulness of our publications. Please take the first available opportunity to fill out this questionnaire and return it to Siemens.

Please do not forget to state the title, order number and release of your manual.

Title of Your Manual: ....

Order No. of Your Manual: .... Release: ....

Please give each of the following questions your own personal mark within the range from 1 (very good) to 5 (poor).

1. Do the contents meet your requirements? ☐
2. Is the information you need easy to find? ☐
3. Is the text easy to understand? ☐
4. Does the level of technical detail meet your requirements? ☐
5. Please rate the quality of the graphics/tables: ☐

Additional comments:

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

Siemens AG
AUT E 1163
Östliche Rheinbrückenstraße 50
D-76181 Karlsruhe
Federal Republic of Germany

From:

Your Name: . . . .

Your Title: . . . .

Company Name: . . . .

Street: . . . .

City, Zip Code: . . . .

Country: . . . .

Phone: . . . .

Please check any industry that applies to you:

- ☐ Automotive
- ☐ Chemical
- ☐ Electrical Machinery
- ☐ Food
- ☐ Instrument and Control
- ☐ Nonelectrical Machinery
- ☐ Pharmaceutical
- ☐ Plastic
- ☐ Pulp and Paper
- ☐ Textiles
- ☐ Transportation
- ☐ Petrochemical

Other . . . .

Remarks Form

Your comments and recommendations will help us to improve the quality and usefulness of our publications. Please take the first available opportunity to fill out this questionnaire and return it to Siemens.

Please do not forget to state the title, order number and release of your manual.

Title of Your Manual: ....
Order No. of Your Manual: .... Release: ....

Please give each of the following questions your own personal mark within the range from 1 (very good) to 5 (poor).

1. Do the contents meet your requirements? ☐
2. Is the information you need easy to find? ☐
3. Is the text easy to understand? ☐
4. Does the level of technical detail meet your requirements? ☐
5. Please rate the quality of the graphics/tables: ☐

Additional comments:

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

Siemens AG
AUT E 1163
Östliche Rheinbrückenstraße 50
D-76181 Karlsruhe
Federal Republic of Germany

From:
Your Name: . . . .
Your Title: . . . .
Company Name: . . . .
Street: . . . .
City, Zip Code: . . . .
Country: . . . .
Phone: . . . .

Please check any industry that applies to you:

- ☐ Automotive
- ☐ Pharmaceutical
- ☐ Chemical
- ☐ Plastic
- ☐ Electrical Machinery
- ☐ Pulp and Paper
- ☐ Food
- ☐ Textiles
- ☐ Instrument and Control
- ☐ Transportation
- ☐ Nonelectrical Machinery
- ☐ Petrochemical

Other . . . .